



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

A Cell-Centered Multiphase ALE Scheme With Structural Coupling

T. A. Dunn

April 16, 2012

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

This work was partially funded by the Joint DoD/DOE Munitions Technology Development Program and the DOE/NNSA Advanced Simulation & Computing (ASC) Program.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

CONTENTS

List of Figures	vi
List of Tables	xviii
Abstract	xxi
Acknowledgments	xxii
1 Introduction	1
1.1 Multiphase Flow Overview	1
1.2 Multiphase Computational Methods	2
1.3 Structural Interaction	5
1.4 Research Objective	7
1.5 Research Contribution and Scope	7
2 Theory	9
2.1 The Multiphase Model	9
2.1.1 Conservation Laws	9
2.1.2 Multiphase Equations	12
2.1.3 DEM Method	15
2.1.4 ALE Equations	17
2.1.5 Axisymmetric Formulation	20
2.1.6 Spatial Discretization	23
2.1.7 Numerical Integration of Terms	24
2.1.7.1 Time Derivative Term	25
2.1.7.2 Convective Term	26
2.1.7.3 Source Term	27
2.1.7.4 Lagrangian/Non-conservative Source Term	27
2.1.8 Continuous Limit of Discrete Equations	29
2.1.9 Time-stepping Scheme	32
2.1.9.1 Time-stepping Stability	33

2.1.10	Geometric Conservation Laws	34
2.1.11	Flux Calculation	37
2.1.12	Boundary Conditions	42
2.1.12.1	Outflow Boundary Conditions	43
2.1.12.2	Wall Boundary Conditions	44
2.1.13	Second-Order Extension	44
2.1.14	Sub-grid Phase Coupling (Relaxation Processes)	48
2.1.14.1	Pressure Relaxation	48
2.1.14.2	Temperature Relaxation	50
2.1.14.3	Velocity Relaxation	53
2.1.14.4	Summary of Relaxation Solution	57
2.2	The Structural Model	59
2.2.1	Lagrangian Equations	60
2.2.2	Lagrangian Algorithm	60
2.3	The Fluid-Structure Interaction Model	62
2.3.1	Stage 1: Fluid to Structure Coupling	62
2.3.2	Stage 2: Fluid Mesh Relaxation	63
2.3.3	Stage 3: Structure to Fluid Coupling	65
3	Results	75
3.1	Single Phase	76
3.1.1	Single-Phase Smooth Test Problem	77
3.1.2	Toro Test Problems	86
3.1.3	Sod Shock Tube	106
3.1.4	Sedov Test	117
3.1.5	Noh Test	144
3.2	Multiple Phases	170
3.2.1	Multiphase Smooth Test Problem	171
3.2.2	Water-Air Shock Tube	186
3.2.3	Shock-Interface Interaction	194

3.2.4	Multiphase Temperature Relaxation	200
3.2.5	Multiphase Nusselt Number	207
3.2.6	Multiphase Velocity Relaxation	217
3.2.7	Multiphase Drag Coefficient	224
3.2.8	Multiphase Drag with Conduction	247
3.2.9	Multiphase Particle Cloud	255
3.2.10	Sommerfeld Dusty Shock Tube	265
3.3	Multiphase-Lagrange Coupling	275
3.3.1	Multiphase Piston	276
3.3.2	Multiphase-Lagrangian Water-Air Shock Tube	282
3.3.3	Lagrange Cantilever Plate Bending	288
3.3.4	Shock Wave Impacting a Steel Panel	295
3.3.5	Multiphase Fluid-Structure Interaction	307
4	Conclusions	312
4.1	Future Work	313
A	Equations of State	317
A.1	Functional Forms	318
A.1.1	Ideal Gas	318
A.1.2	Stiffened Gas	318
A.1.3	JWL	319
A.1.4	Grüneisen	319
B	Drag Coefficient Relations	321
B.1	Newton	321
B.2	Stokes	322
B.3	Schiller and Naumann	322
B.4	Putnam	322
B.5	Ergun	322
B.6	RUC	323

B.7	Gidaspow	323
B.8	Akhatov and Vainshtein	324
C	Nusselt-Number Relations	326
C.1	Constant	326
C.2	Ranz-Marshall	327
D	Solution of the Relaxation Equations	328
D.1	Solving the System of ODE's	328
D.1.1	Analytic Relaxation	329
D.1.2	Backward-Euler Relaxation	336
D.2	Bulk-Gas Properties	339
E	Evaluation of Discretization Error	342

LIST OF FIGURES

2.1	Diagram of computational cells. Zones are notated by capital letters, nodes with numbers, faces by lower case letters.	66
2.2	Diagram of the volume flux through a moving face. The shaded area represents the volume which is swept out as the face moves. This volume is used to satisfy the geometric conservation laws. . . .	67
2.3	An example problem revealing the importance of satisfying the GCL. The colors represent density magnitude.	68
2.4	Characteristic fan for the Riemann problem	69
2.5	Grid stencil used for second-order predictor.	70
2.6	Mesh diagram of the multiphase/Lagrangian-structure interface. The green zones are controlled by the multiphase algorithm; the blue zones are controlled by the staggered-grid FEM.	71
3.1	Numerical results for the single-phase smooth test. The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	81
3.2	Numerical results for the single-phase smooth test. The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	82
3.3	Single-phase numerical results for Toro Test #1. The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	90
3.4	Single-phase numerical results for Toro Test #2. The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	91
3.5	Single-phase numerical results for Toro Test #3. The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	92
3.6	Single-phase numerical results for Toro Test #4. The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	93
3.7	Single-phase numerical results for Toro Test #5. The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	94

3.8	Single-phase numerical results for Toro Test #6. The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	95
3.9	Single-phase numerical results for Toro Test #7. The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	96
3.10	Single-phase numerical results for Toro Test #1. The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	97
3.11	Single-phase numerical results for Toro Test #2. The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	98
3.12	Single-phase numerical results for Toro Test #3. The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	99
3.13	Single-phase numerical results for Toro Test #4. The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	100
3.14	Single-phase numerical results for Toro Test #5. The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	101
3.15	Single-phase numerical results for Toro Test #6. The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	102
3.16	Single-phase numerical results for Toro Test #7. The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	103
3.17	Single-phase numerical results for the Sod Test. The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	110
3.18	Single-phase numerical results for the Sod Test. The tests used an <i>ALE grid</i> with the <i>predictor off</i>	111
3.19	Single-phase numerical results for the Sod Test. The tests used <i>Eulerian/Lagrangian/ALE grids</i> with the <i>predictor off</i>	112
3.20	Single-phase numerical results for the Sod Test. The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	113
3.21	Single-phase numerical results for the Sod Test. The tests used an <i>ALE grid</i> with the <i>predictor on</i>	114

3.22	Single-phase numerical results for the Sod Test. The tests used <i>Eulerian/Lagrangian/ALE grids</i> with the <i>predictor on</i>	115
3.23	Initial mesh geometries for the Sedov tests.	122
3.24	Single-phase numerical results for the Sedov Test #1 (1D Cartesian geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	124
3.25	Single-phase numerical results for the Sedov Test #2 (1D Cylindrical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	125
3.26	Single-phase numerical results for the Sedov Test #3 (1D Spherical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	126
3.27	Single-phase numerical results for the Sedov Test #4 (1D-axisymmetric Cylindrical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	127
3.28	Single-phase numerical results for the Sedov Test #5 (2D Cylindrical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	128
3.29	Single-phase numerical results for the Sedov Test #6 (2D-axisymmetric Spherical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	129
3.30	Single-phase numerical results for the Sedov Test #7 (3D Spherical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	130
3.31	Single-phase numerical results for the Sedov Test #1 (1D Cartesian geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	131
3.32	Single-phase numerical results for the Sedov Test #2 (1D Cylindrical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	132
3.33	Single-phase numerical results for the Sedov Test #3 (1D Spherical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	133

3.34	Single-phase numerical results for the Sedov Test #4 (1D-axisymmetric Cylindrical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	134
3.35	Single-phase numerical results for the Sedov Test #5 (2D Cylindrical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	135
3.36	Single-phase numerical results for the Sedov Test #6 (2D-axisymmetric Spherical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	136
3.37	Single-phase numerical results for the Sedov Test #7 (3D Spherical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	137
3.38	Single-phase numerical scatter-plot results for the Sedov Test #5 (2D Cylindrical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	138
3.39	Single-phase numerical scatter-plot results for the Sedov Test #6 (2D-axisymmetric Spherical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	139
3.40	Single-phase numerical scatter-plot results for the Sedov Test #7 (3D Spherical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	140
3.41	Single-phase numerical scatter-plot results for the Sedov Test #5 (2D Cylindrical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	141
3.42	Single-phase numerical scatter-plot results for the Sedov Test #6 (2D-axisymmetric Spherical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	142
3.43	Single-phase numerical scatter-plot results for the Sedov Test #7 (3D Spherical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	143

3.44	Initial mesh geometries for the Noh tests.	148
3.45	Single-phase numerical results for the Noh Test #1 (1D Cartesian geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor off</i> . .	150
3.46	Single-phase numerical results for the Noh Test #2 (1D Cylindrical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor off</i> . .	151
3.47	Single-phase numerical results for the Noh Test #3 (1D Spherical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor off</i> . .	152
3.48	Single-phase numerical results for the Noh Test #4 (1D-axisymmetric Cylindrical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	153
3.49	Single-phase numerical results for the Noh Test #5 (2D Cylindrical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor off</i> . .	154
3.50	Single-phase numerical results for the Noh Test #6 (2D-axisymmetric Spherical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	155
3.51	Single-phase numerical results for the Noh Test #7 (3D Spherical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor off</i> . .	156
3.52	Single-phase numerical results for the Noh Test #1 (1D Cartesian geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor on</i> . .	157
3.53	Single-phase numerical results for the Noh Test #2 (1D Cylindrical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor on</i> . .	158
3.54	Single-phase numerical results for the Noh Test #3 (1D Spherical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor on</i> . .	159
3.55	Single-phase numerical results for the Noh Test #4 (1D-axisymmetric Cylindrical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	160
3.56	Single-phase numerical results for the Noh Test #5 (2D Cylindrical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor on</i> . .	161

3.57	Single-phase numerical results for the Noh Test #6 (2D-axisymmetric Spherical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	162
3.58	Single-phase numerical results for the Noh Test #7 (3D Spherical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	163
3.59	Single-phase numerical scatter-plot results for the Noh Test #5 (2D Cylindrical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	164
3.60	Single-phase numerical scatter-plot results for the Noh Test #6 (2D-axisymmetric Spherical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	165
3.61	Single-phase numerical scatter-plot results for the Noh Test #7 (3D Spherical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	166
3.62	Single-phase numerical scatter-plot results for the Noh Test #5 (2D Cylindrical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	167
3.63	Single-phase numerical scatter-plot results for the Noh Test #6 (2D-axisymmetric Spherical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	168
3.64	Single-phase numerical scatter-plot results for the Noh Test #7 (3D Spherical geometry). The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	169
3.65	Numerical results for the multiphase smooth test. The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	176
3.66	Numerical results for the multiphase smooth test. The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	178
3.67	Multiphase numerical results for the water-air shock-tube test. The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	189

3.68	Multiphase numerical results for the water-air shock-tube test. The tests used an <i>ALE grid</i> with the <i>predictor off</i>	190
3.69	Multiphase numerical results for the water-air shock-tube test. The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	191
3.70	Multiphase numerical results for the water-air shock-tube test. The tests used an <i>ALE grid</i> with the <i>predictor on</i>	192
3.71	Multiphase numerical results for the shock-interface test. The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	197
3.72	Multiphase numerical results for the shock-interface test. The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	198
3.73	Multiphase numerical results for the temperature-relaxation test. Phase temperatures were computed using the backward-Euler method. The higher temperature curves represent the solid (phase <i>b</i>) and the lower temperature curves represent the gas (phase <i>a</i>). . .	203
3.74	Multiphase numerical results for the temperature-relaxation test. Phase temperatures were computed using the analytical method. The higher temperature curves represent the solid (phase <i>b</i>) and the lower temperature curves represent the gas (phase <i>a</i>).	204
3.75	Multiphase numerical results for the Nusselt-number tests. The Nusselt-number law used a <i>fixed relaxation constant</i> . The higher temperature curves represent the solid (phase <i>b</i>) and the lower temperature curves represent the gas (phase <i>a</i>).	211
3.76	Multiphase numerical results for the Nusselt-number tests. The Nusselt-number law used a <i>constant Nusselt number</i> . The higher temperature curves represent the solid (phase <i>b</i>) and the lower temperature curves represent the gas (phase <i>a</i>).	212

3.77	Multiphase numerical results for the Nusselt-number tests. The Nusselt-number law used a <i>Ranz-Marshall correlation</i> . The higher temperature curves represent the solid (phase <i>b</i>) and the lower temperature curves represent the gas (phase <i>a</i>).	213
3.78	Multiphase numerical results for the velocity-relaxation test. Phase velocities were computed using the backward-Euler method. The higher velocity curves represent the solid (phase <i>b</i>) and the lower velocity curves represent the gas (phase <i>a</i>).	220
3.79	Multiphase numerical results for the velocity-relaxation test. Phase velocities were computed using the analytical method. The higher velocity curves represent the solid (phase <i>b</i>) and the lower velocity curves represent the gas (phase <i>a</i>).	221
3.80	Multiphase numerical results for the drag-coefficient tests. The drag-coefficient law used a <i>fixed relaxation constant</i> . The higher velocity curves represent the solid (phase <i>b</i>) and the lower velocity curves represent the gas (phase <i>a</i>).	229
3.81	Multiphase numerical results for the drag-coefficient tests. The drag-coefficient law used a <i>constant drag coefficient</i> . The higher velocity curves represent the solid (phase <i>b</i>) and the lower velocity curves represent the gas (phase <i>a</i>).	230
3.82	Multiphase numerical results for the drag-coefficient tests. The drag-coefficient law used a <i>Stokes correlation</i> . The higher velocity curves represent the solid (phase <i>b</i>) and the lower velocity curves represent the gas (phase <i>a</i>).	231
3.83	Multiphase numerical results for the drag-coefficient tests. The drag-coefficient law used a <i>Schiller correlation</i> . The higher velocity curves represent the solid (phase <i>b</i>) and the lower velocity curves represent the gas (phase <i>a</i>).	232

3.84	Multiphase numerical results for the drag-coefficient tests. The drag-coefficient law used a <i>Putnam correlation</i> . The higher velocity curves represent the solid (phase <i>b</i>) and the lower velocity curves represent the gas (phase <i>a</i>).	233
3.85	Multiphase numerical results for the drag-coefficient tests. The drag-coefficient law used a <i>Ergun correlation</i> . The higher velocity curves represent the solid (phase <i>b</i>) and the lower velocity curves represent the gas (phase <i>a</i>).	234
3.86	Multiphase numerical results for the drag-coefficient tests. The drag-coefficient law used a <i>RUC correlation</i> . The higher velocity curves represent the solid (phase <i>b</i>) and the lower velocity curves represent the gas (phase <i>a</i>).	235
3.87	Multiphase numerical results for the drag-coefficient tests. The drag-coefficient law used a <i>Gidaspow correlation</i> . The higher velocity curves represent the solid (phase <i>b</i>) and the lower velocity curves represent the gas (phase <i>a</i>).	236
3.88	Multiphase numerical results for the drag-coefficient tests. The drag-coefficient law used a <i>Akhatov correlation</i> . The higher velocity curves represent the solid (phase <i>b</i>) and the lower velocity curves represent the gas (phase <i>a</i>).	237
3.89	Multiphase numerical results for the coupled drag/conduction tests: velocity results. The higher velocity curves represent the solid (phase <i>b</i>) and the lower velocity curves represent the gas (phase <i>a</i>).	251
3.90	Multiphase numerical results for the coupled drag/conduction tests: temperature results. The higher temperature curves represent the solid (phase <i>b</i>) and the lower temperature curves represent the gas (phase <i>a</i>).	252

3.91	Multiphase numerical results for the position of the center of mass of a particle cloud. The tests used an <i>Eulerian grid</i> with the <i>predictor off</i> .	259
3.92	Multiphase numerical results for the velocity of the center of mass of a particle cloud. The tests used an <i>Eulerian grid</i> with the <i>predictor off</i> .	260
3.93	Multiphase numerical results for the volume fraction of a particle cloud. The tests used an <i>Eulerian grid</i> with the <i>predictor off</i> .	261
3.94	Multiphase numerical results for the position of the center of mass of a particle cloud. The tests used an <i>Eulerian grid</i> with the <i>predictor on</i> .	262
3.95	Multiphase numerical results for the velocity of the center of mass of a particle cloud. The tests used an <i>Eulerian grid</i> with the <i>predictor on</i> .	263
3.96	Multiphase numerical results for the volume fraction of a particle cloud. The tests used an <i>Eulerian grid</i> with the <i>predictor on</i> .	264
3.97	Sommerfeld Shock Tube Setup. (Not shown to scale)	270
3.98	Multiphase numerical results for the Sommerfeld shock tube. The shock Mach number is plotted. The tests used an <i>Eulerian grid</i> with the <i>predictor off</i> .	271
3.99	Multiphase numerical results for the Sommerfeld shock tube. The shock Mach number is plotted. The tests used an <i>Eulerian grid</i> with the <i>predictor on</i> .	272
3.100	Multiphase numerical results for the Sommerfeld shock tube. The shock Mach number is plotted. The tests used an <i>ALE grid</i> with the <i>predictor off</i> .	273
3.101	Multiphase numerical results for the Sommerfeld shock tube. The shock Mach number is plotted. The tests used an <i>ALE grid</i> with the <i>predictor on</i> .	274

3.102	Initial geometry for the multiphase piston test. Region 1 is shown in red and contains a multiphase mixture. Region 2 is shown in green and contains a solid material. Region 3 is shown in blue and contains a single-phase gas.	279
3.103	Numerical results for the position of the center of mass of a rigid piston.	280
3.104	Numerical results for the velocity of the center of mass of a rigid piston.	281
3.105	Numerical results for the multiphase-Lagrangian water-air shock-tube test. The tests were run with the <i>predictor off</i>	285
3.106	Numerical results for the multiphase-Lagrangian water-air shock-tube test. The tests were run with the <i>predictor on</i>	286
3.107	Bending-plate problem setup.	292
3.108	Numerical results for the bending-plate problem. Plate displacement is plotted.	293
3.109	Problem setup for a shock wave impacting a steel panel. Figure taken from [37].	300
3.110	Numerical results for a shock wave impacting a steel panel. Experimental and Numerical Shadowgraphs are shown at various time levels. Left column is the experimental shadowgraphs of Giordano. Center column is the numerical calculation of Giordano. Right column is the current calculation on Grid x5.	301
3.111	Numerical results for a shock wave impacting a steel panel. Shadowgraph taken at $t = 420 \mu s$. From top to bottom: Giordano Experiment, Giordano Calculation, Current Calculation on Grid x3, Grid x5, Grid x9, Grid x17.	304
3.112	Numerical results for a shock wave impacting a steel panel. Plot of panel displacement.	305

3.113	Numerical results for a shock wave impacting a steel panel. Plot of pressure probe data.	306
3.114	Numerical results for a multiphase fluid impacting a steel panel. Plot of panel displacement.	310
B.1	Comparison of experimental data to empirical relations for the drag coefficient of a smooth solid sphere. Experimental data taken from White [41].	325

LIST OF TABLES

2.1	Definitions of surface contacts used for computing the convective and Lagrangian fluxes.	72
2.2	Definitions of inter-cell phase function used for computing the convective fluxes.	73
2.3	Definitions of inter-cell jump functions used for computing the Lagrangian fluxes.	74
3.1	Numerical error and convergence rates for the single-phase smooth test. The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	83
3.2	Numerical error and convergence rates for the single-phase smooth test. The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	84
3.3	Numerical error and convergence rates for the single-phase smooth test. The tests used an <i>Eulerian grid</i> with the <i>predictor on/no limiter</i>	85
3.4	Position of discontinuity and end time for the Toro test problems.	104
3.5	Initial conditions for the Toro test problems.	105
3.6	Initial conditions for the Sod test problem.	116
3.7	Numerical error and convergence rates for the multiphase smooth test. The tests used an <i>Eulerian grid</i> with the <i>predictor off</i>	180
3.8	Numerical error and convergence rates for the multiphase smooth test. The tests used an <i>Eulerian grid</i> with the <i>predictor on</i>	182
3.9	Numerical error and convergence rates for the multiphase smooth test. The tests used an <i>Eulerian grid</i> with the <i>predictor on/no limiter</i>	184
3.10	Initial conditions for the water-air shock-tube problem	193
3.11	Initial conditions for the shock-interface test problem.	199
3.12	Numerical error and convergence rates for the temperature-relaxation test. Phase temperatures were computed using the backward-Euler method.	205

3.13	Numerical error and convergence rates for the temperature-relaxation test. Phase temperatures were computed using the analytical method.	206
3.14	Numerical error and convergence rates for the Nusselt-number tests. The Nusselt-number law used a <i>fixed relaxation constant</i>	214
3.15	Numerical error and convergence rates for the Nusselt-number tests. The Nusselt-number law used a <i>constant Nusselt number</i>	215
3.16	Numerical error and convergence rates for the Nusselt-number tests. The Nusselt-number law used a <i>Ranz-Marshall correlation</i>	216
3.17	Numerical error and convergence rates for the velocity-relaxation test. Phase velocities were computed using the backward-Euler method.	222
3.18	Numerical error and convergence rates for the velocity-relaxation test. Phase velocities were computed using the analytical method.	223
3.19	Numerical error and convergence rates for the drag-coefficient tests. The drag-coefficient law used a <i>fixed relaxation constant</i>	238
3.20	Numerical error and convergence rates for the drag-coefficient tests. The drag-coefficient law used a <i>constant drag coefficient</i>	239
3.21	Numerical error and convergence rates for the drag-coefficient tests. The drag-coefficient law used a <i>Stokes correlation</i>	240
3.22	Numerical error and convergence rates for the drag-coefficient tests. The drag-coefficient law used a <i>Schiller correlation</i>	241
3.23	Numerical error and convergence rates for the drag-coefficient tests. The drag-coefficient law used a <i>Putnam correlation</i>	242
3.24	Numerical error and convergence rates for the drag-coefficient tests. The drag-coefficient law used a <i>Ergun correlation</i>	243
3.25	Numerical error and convergence rates for the drag-coefficient tests. The drag-coefficient law used a <i>RUC correlation</i>	244

3.26	Numerical error and convergence rates for the drag-coefficient tests. The drag-coefficient law used a <i>Gidaspow correlation</i>	245
3.27	Numerical error and convergence rates for the drag-coefficient tests. The drag-coefficient law used a <i>Akhatov correlation</i>	246
3.28	Numerical error and convergence rates for the coupled drag/conduction tests: velocity results.	253
3.29	Numerical error and convergence rates for the coupled drag/conduction tests: temperature results.	254
3.30	Initial conditions for the multiphase-Lagrangian water-air shock- tube problem.	287
3.31	Numerical results for the bending-plate problem. Theoretical and computed plate displacements and periods are tabulated.	294
3.32	Numerical results for a multiphase fluid impacting a steel panel. Table of maximum panel displacement.	311

ABSTRACT

A novel computational scheme has been developed for simulating compressible multiphase flows interacting with solid structures. The multiphase fluid is computed using a Godunov-type finite-volume method. This has been extended to allow computations on moving meshes using a direct arbitrary-Eulerian-Lagrangian (ALE) scheme. The method has been implemented within a Lagrangian hydrocode, which allows modeling the interaction with Lagrangian structural regions. Although the above scheme is general enough for use on many applications, the ultimate goal of the research is the simulation of heterogeneous energetic material, such as explosives or propellants. The method is powerful enough for application to all stages of the problem, including the initial burning of the material, the propagation of blast waves, and interaction with surrounding structures. The method has been tested on a number of canonical multiphase tests as well as fluid-structure interaction problems.

ACKNOWLEDGMENTS

The completion of this work has been a long time coming. It would not have been possible without the help and encouragement of many people over the past few years.

I would like to thank my thesis advisor, Professor Wolfgang Kollmann. I am very grateful for his many contributions during the course of my research. I am particularly indebted to him for continuing to provide guidance even after his retirement. I would also like to thank my dissertation committee, consisting of Professor Jean-Jacques Chattot and Professor Ian Kennedy. Their feedback was very helpful for the completion of this manuscript.

I acknowledge the support of Lawrence Livermore National Laboratory (LLNL). I thank the Laboratory management for supporting me financially throughout the course of my degree. I also thank my project managers for providing stimulating assignments with enough flexibility to allow me to apply my work towards my thesis research. This research would not have been possible without the use of LLNL computers, office space, and library.

I acknowledge the support of my LLNL co-workers. I would like to thank Dr. Rose McCallen for encouraging me to pursue this degree, as well as providing technical and managerial guidance throughout my career at LLNL. This thesis would not have been possible without the technical expertise of Dr. David Stevens. Dave was responsible for the initial implementation of the multiphase model used in this study, and many of the methods used in this study were a result of his groundwork. His knowledge of multiphase flow and numerical algorithms were highly valuable. I thank the entire *downstairs* code team for helping me with all sorts of computer science and physics related questions.

I would like to thank my family. My mom and dad have provided love and encouragement through all the years I've spent in school. Finally, I thank my beautiful wife, Erika, for her love and encouragement (and patience) as I finished writing this dissertation.

Chapter 1

Introduction

The reaction and propagation of energetic materials has been the subject of extensive research. Recently, the importance of the multi-component characteristics of these materials has been recognized, leading to greater understanding of areas such as deflagration to detonation transition (DDT) and the performance of non-ideal explosives and propellants. The desire to accurately model these phenomena has resulted in a need for new computational models capable of efficiently simulating these multi-component mixtures and the interactions with their surroundings.

1.1 Multiphase Flow Overview

Interest in the dynamics of multi-material fluid mixtures has grown recently. This subcategory of fluid mechanics is often referred to as multiphase or multi-component flow in the literature and covers a wide spectrum of flow conditions and applications. Multiphase flows are categorized by phase of the materials: gas-liquid, gas-solid, or liquid-solid flow. The current research is most concerned with particle-laden gas flow, where discrete solid particles are embedded in a carrier gas. The presence of inert or reactive particles can significantly affect the flow through momentum and energy exchanges. These types of flows are encountered in many engineering applications, such as solid-propellant rocket motors, blast waves in dusty atmospheres, and gas-permeable reactive granular materials.

Gas-particle flows can be further categorized into flow regimes according to the particle concentration of the flow. Flows with low particle concentrations are referred to as *dilute* multiphase flow. Here, the particles have little effect on the carrier fluid and the particle motion is generally dominated by the interaction with the surrounding fluid, as opposed to the presence of other particles. As the particle concentration increases, the particles may start to contribute to the dynamics of the gas through momentum and energy transfer. At still larger concentrations, the volume occupied by the particles begins to influence the fluid flow. In addition, as particles become close in proximity to each other, particles will start to influence neighboring particles through wake effects or even particle-particle collisions. These later phenomena are all characteristics of the *dense* multiphase regime.

1.2 Multiphase Computational Methods

Computational simulation is an integral part of engineering analysis and can help to provide greater insight into complex physical processes. Hence, modeling multiphase gas-particle flow has become an important area of computational mechanics with many unique characteristics and challenges. The literature provides a number of reviews on the current methodologies [1–3] as well as a number of textbooks on the subject [4]. The different computational approaches vary widely in their prediction fidelity and computational requirements. Many of the models incorporate various simplifying assumptions, thereby limiting their use to particular multiphase regimes. Thus, the numerical algorithms used for simulation of multiphase flow require careful consideration of the flow regimes along with the relevant numerics.

As the capacity of computational resources has improved, so has the ability to perform more challenging simulations. Focus has shifted from running isolated physics models to performing full-system calculations. With respect to the multiphase modeling of energetic materials, this includes the initial burning of

the material, the propagation of blast waves, and interaction with surrounding structures. This calculation encompasses the full range of multiphase flow regimes. Initially, during the detonation phase, the granular multiphase mixture is characteristic of a very dense porous flow regime, with the solid phase occupying most of the volume. As the blast wave propagates, the flow becomes more dilute with discrete particles being controlled primarily by aerodynamic forces. In between there is a transition from dense to dilute flow where the material is accelerated and dispersed. Therefore, the challenge for the current research is to develop a numerical scheme valid for all of these regimes.

One option, which is common in combustion simulations, is to derive conservation equations for averaged flow variables. This is the single-phase option and neglects the discrete nature of the separate phases. This method could homogenize the phases together or keep track of individual species for evaluation of the state data. Although this is a good choice for some problems, this method produces erroneous results for phases that do not have similar velocities.

Another option is to resolve all the material interfaces in the problem. This is also known as a multi-fluid algorithm in which each material interface is explicitly tracked through the simulation [5–7]. This is only feasible for problems where the size of the interfaces are on the same order as the length scale of the overall problem. For example, this is practical for simulating the dynamics of a single liquid droplet, or the interaction of a small number of particles. However, tracking millions of micron sized particles as they travel several meters is not practical with current computational resources.

For dilute particulate flows, a continuum based Eulerian treatment is often used for the gas phase, while the discrete particles are individually tracked in a Lagrange methodology. Lagrangian particle-tracking algorithms have been used for compressible gas-particle flows by a number of researchers [8–11]. These methods have the advantage of tracking the evolution of individual particles more accurately than Eulerian methods. However, for large numbers of particles this can be very

computationally intensive.

Another option is to treat both the gas and particle phases as separate continuum materials and use an Eulerian approach for both. The premise of this approach is that each component of the multiphase mixture is treated as an independent continuum material with its own momentum and thermodynamic state data. Therefore, separate field equations are derived for each component assuming the coexistence of the phases at every point in the flow-field. The particle material interfaces are assumed to be much smaller than the grid resolution in the problem. Therefore, their interactions are modeled using sub-grid constitutive relations. This method allows for more accurate inter-phase coupling and is more computationally efficient for large numbers of particles. However, there is no universally accepted formulation of the conservation laws governing these complex flows. Thus, several different approaches have been employed for various multiphase problems.

Loth and his coworkers [9, 12–14] looked at shock propagation and detonations with inert particles. They used an unstructured finite-element method using an Eulerian-Eulerian description of the dilute two-phase flow. However, their hydrodynamic model neglects the volume occupied by the particles. This assumption is not always valid, particularly in the granular bed of an energetic material.

Papalexandris [15] modeled detonations of gas-particle mixtures taking the particle volume into account. However, the solid phase was assumed to be incompressible, ignoring the effect of solid phase pressure and neglecting the non-conservative *nozzling* terms. This could have an adverse effect on the compaction of the solid material. Similarly, Powers et al. [16, 17] neglected the nozzling terms, but did use a compressible solid.

The pioneering work of Baer and Nunziato [18] used mixture theory to investigate convective burning in DDT. Their formulation (commonly referred to as BN in the literature) takes into account the compressibility of all phases. This

model includes the interfacial nozzling terms in the formulation. This raises the question of how to evaluate these terms and can lead to non-unique estimation of the interface pressure and velocity. The choice made for these terms determines the differential expression of the pressure-equilibrium condition. See, for example, Saurel and Abgrall [19] for an alternative representation.

Abgrall and Saurel [20] used the averaging techniques of Drew and Passman [21] to derive conservation laws at the discrete level. They referred to this technique as the Discrete Equation Method (DEM). This approach applies a finite-volume scheme to a multiphase control volume, obtaining a Riemann solution at all boundaries and internal interfaces. The method results in a system of multi-component conservation laws which are unconditionally hyperbolic. The continuous limit of the discretization results in explicit expressions for the non-conservative terms and relaxation processes. The overall assumptions of this method are directly applicable to simulating reactive granular materials, as seen in Chinnayya et al. [22].

1.3 Structural Interaction

A particular interest, when studying energetic materials, is their interaction with nearby structural material. The energetic material is often confined by other inert material. The reaction strongly depends on the various pressure waves propagating through the reactive mixture and surrounding material. In the far-field, the effects of energetic materials on external structures is of great interest. Accurately determining the damage to nearby infrastructure has many applications to defense, security, and safety programs.

One computational strategy for coupling the interactions of fluids and structures together is to solve both materials on a moving grid. There are a number of advantages to solving flow problems on moving meshes. These methods are often referred to as arbitrary-Eulerian-Lagrangian (ALE) methods. These algorithms allow the solution of problems with moving boundaries. In

addition, mesh points can be rearranged during the solution to better resolve specific flow features, such as shocks and material interfaces. There are essentially two approaches for solving the ALE equations: a split approach and an un-split approach. The split approach separates the time-step into a Lagrangian step followed by mesh relaxation and material advection or remap step [7, 23–26]. Although the Lagrange plus remap approach has some benefits for single-phase hydrodynamic problems, its application to the multiphase conservation laws is not straightforward. This is because each phase has its own distinct velocity and there is no single Lagrangian motion of the material. Thus, the multiphase equations are inherently Eulerian in nature. Ruggirello et al. [27] used the bulk material motion to formulate a multiphase Lagrangian method, but this requires the use of a *drift* velocity, such that the material is not constrained to the computational cell as in a standard single-phase Lagrangian method. The un-split method, also referred to as a direct-ALE method, reformulates the governing equations in a moving reference frame and solves the grid motion and fluid flow simultaneously in a single step [26, 28–35]. Saurel et al. [36] applied the DEM on moving meshes using the un-split direct-ALE approach. However, the method was restricted to one-dimensional rectangular (Cartesian) geometries. Due to the apparent Cartesian grid structure of the underlying formulation, the algorithms do not directly extend to multiple dimensions.

There are many examples of single-phase fluid algorithms coupled to structural dynamics codes to investigate fluid-structure interaction in the literature, e.g. [37]. However, computational investigations on the interaction of multiphase fluids with structures are rare. Frost et al. [38] studied the explosive dispersal of solid particles. A loosely-coupled technique was used to evaluate the response of a cantilever rod to the multiphase mixture. The BN multiphase model was used to compute the momentum and pressure of the multiphase fluid. This was then used to compute time-dependent force boundary conditions for a commercial structural response code. This loosely coupled approach has a number of deficiencies when the motion

of the fluid and solid are strongly coupled together. Knap and Stevens [39] used the DEM to model structural response using a diffuse boundary. In this method a single Eulerian mesh was used to compute both the fluid and solid response. The solid was treated as a distinct phase in the multiphase model. There was no interface tracking employed, resulting in a diffusive interface between the fluid and solid. It was found that excessively fine grid resolution was required to resolve the boundary adequately. It is expected that explicitly tracking the Lagrangian motion of the structural boundary would lead to improved representation of the structural response. However, the author knows of no currently available reports describing an Eulerian-Eulerian sub-grid multiphase solver capable of interacting with a Lagrangian structure.

1.4 Research Objective

The objective of the current research is to extend the multiphase DEM method for use in fluid-structure interaction calculations. The governing equations have been reformulated using a direct-ALE scheme and implemented within a Lagrangian hydrocode, allowing interaction with Lagrangian structural regions. The primary application of the model is the simulation of energetic material. The method is powerful enough for application to all stages of the problem, including the initial burning of the material, the propagation of blast waves, and interaction with surrounding structures. Although the eventual goal is to utilize the model for reactive materials, the focus so far has been on evaluating the method using inert materials. The method is applicable to unstructured meshes in two and three dimensions.

1.5 Research Contribution and Scope

The computational model discussed in this report has been implemented in an existing hydrocode. A description of this code is found in [40]. The code is designed to run coupled multi-physics problems with the capability to model explicit and implicit structural mechanics, heat transfer, chemistry, MHD, and fluid

flow. It also allows material advection for running on ALE (moving) meshes. The software is currently under development with a large number of active developers. Implementing the current multiphase model into an existing framework had many advantages as well as some complications.

One advantage was that the code infrastructure was already in place. This included a number of useful data structures for holding and manipulating data. Also, routines for input-file parsing, mesh generation, data output, and post-processing were already written. The underlying code automatically generates most of the geometric information, such as element connectivity, cell volumes, surface normals, and volume fluxes. The infrastructure for performing parallel communication was also provided, resulting the ability to run the current model on massively parallel computer architectures.

Another advantage was that a number of physics models were written before this project started. For example, the Lagrangian structural solver used in the current study was already available in the code, thus it did not need to be developed specifically for this project. Similarly, the mesh-relaxation algorithms used to move the mesh were in the code. Existing routines to evaluate equations of state and other material parameters were also reused. Therefore, in subsequent derivations of the method, the preexisting capabilities will only be discussed briefly and specific details of their methodologies will be omitted.

Before, the current project started, a preliminary multiphase capability had also been developed. This model was added in order to evaluate a number of different models. One of the models was the DEM, which is used in the current study. That model has been largely rewritten for the current study. The model was made more robust by fixing bugs and making all of the models work together more appropriately. The most relevant enhancements for the current study was the addition of the ALE capability to run on moving meshes and adding the coupling to the Lagrangian structure.

Chapter 2

Theory

A novel multiphase model has been developed for the simulation of multiphase mixtures interacting with solid structures. This chapter derives this model and discusses its implementation. First, a detailed description of the multiphase model is provided with particular emphasis on its extension to moving-mesh calculations. Next, there is a brief description of the Lagrangian structural model. This is followed with details of how it was coupled to the new multiphase capability.

2.1 The Multiphase Model

This section derives the multiphase model. It begins with a general discussion of the equations relevant to multiphase mixtures. These are used to derive the hydrodynamic conservation laws used in the current study. It is then shown how these are discretized and solved numerically. The section finishes with a description of the sub-grid relaxation models used in the current formulation.

2.1.1 Conservation Laws

The flow of a bulk material is described by a system of conservation laws commonly referred to as the Euler equations. The derivation of these equations is well known and can be found in various texts on continuum mechanics (e.g. [41–43]). For a fixed Eulerian coordinate frame, $\vec{x} \in \mathbb{R}^3$, these equations are given as

Mass Balance (*Continuity Equation*)

$$\frac{\partial(\rho)}{\partial t} + \frac{\partial(\rho u_j)}{\partial x_j} = 0 \quad (2.1)$$

Momentum Balance (*Newton's Second Law*)

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial(\rho u_i u_j - \sigma_{ji})}{\partial x_j} = 0 \quad (2.2)$$

Energy Balance (*First Law of Thermodynamics*)

$$\frac{\partial(\rho E)}{\partial t} + \frac{\partial(\rho u_j E - \sigma_{ji} u_i)}{\partial x_j} = 0 \quad (2.3)$$

which represent the conservation of mass (2.1), momentum (2.2), and energy (2.3). They are presented in conservative form and make use of index notation where summation is implied over repeated indices. The above equations neglect a number of physical effects, such as body force, viscosity, and heat flux. Therefore, the Froude, Reynolds, and Nusselt numbers are all assumed to be large with respect to the length scales associated with the bulk flow. However, as will be shown in subsequent sections, this does not prevent the model from accounting for viscous forces and heat transfer between phases at the sub-grid level.

In the above equations, ρ denotes the mass density of the material and u_i is the i component of the velocity vector. The specific total energy, E , is given by

$$E = \varepsilon + \frac{1}{2} u_j u_j$$

where ε is the specific internal energy. The Cauchy (total) stress tensor, $\boldsymbol{\sigma}$, is symmetric ($\sigma_{ij} = \sigma_{ji}$) and can be expanded as

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}$$

The stress is decomposed into the normal and deviatoric components such that

$$\sigma_{ij} = -p\delta_{ij} + \tau_{ij}$$

where the pressure obeys an appropriate equation of state ($p = p(\rho, \varepsilon)$) and the deviatoric stress ($\boldsymbol{\tau}$) is obtained from a constitutive relation dependent upon the motion of the material. Although the temperature does not appear in the above equations, it is common to include a caloric equation of state to compute it ($T = T(\rho, \varepsilon)$). For materials with strength, additional equations must be added to the conservation laws to transport elastic and plastic deformation histories (see [44–46]). The current work will neglect these equations and focus on materials with no strength ($\boldsymbol{\tau} = 0$), while continuing to use the full stress tensor in the notation.

In subsequent derivations, it will be convenient to represent the conservation laws with a more compact notation. Using tensor notation (2.1–2.3) can be expressed as

$$\left. \frac{\partial \mathbf{W}(\vec{x}, t)}{\partial t} \right|_{\vec{x}} + \left. \frac{\partial \mathbf{H}_i(\vec{x}, t)}{\partial x_i} \right|_t = \mathbf{S}(\vec{x}, t) \quad (2.4)$$

where \mathbf{W} is the vector of conserved quantities

$$\mathbf{W}(\vec{x}, t) = \begin{bmatrix} 1 \\ \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{bmatrix}$$

and the flux, \mathbf{H}_i , refers to a vector given by row i of tensor \mathbf{H}

$$\mathbf{H}(\vec{x}, t) = \begin{bmatrix} 0 & \rho u_1 & \rho u_1 u_1 - \sigma_{11} & \rho u_1 u_2 - \sigma_{21} & \rho u_1 u_3 - \sigma_{31} & \rho u_1 E - \sigma_{1j} u_j \\ 0 & \rho u_2 & \rho u_2 u_1 - \sigma_{12} & \rho u_2 u_2 - \sigma_{22} & \rho u_2 u_3 - \sigma_{32} & \rho u_2 E - \sigma_{2j} u_j \\ 0 & \rho u_3 & \rho u_3 u_1 - \sigma_{13} & \rho u_3 u_2 - \sigma_{23} & \rho u_3 u_3 - \sigma_{33} & \rho u_3 E - \sigma_{3j} u_j \end{bmatrix}$$

The flux can be further decomposed into its convective and Lagrangian components, such that

$$\mathbf{H}_i(\vec{x}, t) = u_i \mathbf{W} + \mathbf{P}_i$$

where the Lagrangian flux, \mathbf{P} , is defined as

$$\mathbf{P}(\vec{x}, t) = \begin{bmatrix} -u_1 & 0 & -\sigma_{11} & -\sigma_{21} & -\sigma_{31} & -\sigma_{1j}u_j \\ -u_2 & 0 & -\sigma_{12} & -\sigma_{22} & -\sigma_{32} & -\sigma_{2j}u_j \\ -u_3 & 0 & -\sigma_{13} & -\sigma_{23} & -\sigma_{33} & -\sigma_{3j}u_j \end{bmatrix}$$

and the repeated indices (j) in the energy column imply summation. A generic source term has been added to the above system, where

$$\mathbf{S}(\vec{x}, t) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

For the Euler equations presented in this section, the source term is zero. However, the source is included here to facilitate inclusion of source terms later, such as body force or chemical reaction terms.

Notice that the first row in (2.4) represents the trivial identity

$$\frac{\partial 1}{\partial t} + \frac{\partial 0_j}{\partial x_j} = 0$$

This is included to simplify the addition of the volume-fraction evolution in later sections.

2.1.2 Multiphase Equations

The Euler equations presented in the previous section describe the bulk motion of a single material. The current work, however, is interested in the motion of multi-component mixtures. These flows are characterized by immiscible materials separated by discrete interfaces. In principle, it is possible to apply the Euler equations (2.1–2.3) to each component separately up to the interface, but not across it. Special treatment is required to account for the discontinuities at the material boundary. This can be done through jump conditions across the interface

to account for mass, momentum, and energy exchange between the materials. These terms can be complicated and require detailed knowledge of the surface geometry.

Modeling multi-component mixtures with many interfaces can become impractical. Therefore, it is desirable to model these interface interactions in an average sense. The most general form of the averaged multiphase flow equations is the non-equilibrium model. In this model each phase has its own velocity and thermodynamic state (pressure, temperature, etc.). The compressibility of each phase is accounted for in this model. Considering two-phase gas-particle flow, this model is described by a system of seven partial differential equations: conservation equations for mass, momentum, and energy for each phase, plus an additional equation for the evolution of volume fraction. For one-dimensional flow, these equations are given as

$$\begin{aligned}
\frac{\partial(\alpha_g)}{\partial t} &= -u^I \frac{\partial(\alpha_g)}{\partial x} - \mu(p_p - p_g) \\
\frac{\partial(\alpha_g \rho_g)}{\partial t} + \frac{\partial(\alpha_g \rho_g u_g)}{\partial x} &= 0 \\
\frac{\partial(\alpha_g \rho_g u_g)}{\partial t} + \frac{\partial(\alpha_g \rho_g u_g u_g + \alpha_g p_g)}{\partial x} &= p^I \frac{\partial(\alpha_g)}{\partial x} + \lambda(u_p - u_g) \\
\frac{\partial(\alpha_g \rho_g E_g)}{\partial t} + \frac{\partial(\alpha_g \rho_g E_g u_g + \alpha_g p_g u_g)}{\partial x} &= p^I u^I \frac{\partial(\alpha_g)}{\partial x} + \mu p^I (p_p - p_g) + \lambda u^I (u_p - u_g) \quad (2.5) \\
\frac{\partial(\alpha_p \rho_p)}{\partial t} + \frac{\partial(\alpha_p \rho_p u_p)}{\partial x} &= 0 \\
\frac{\partial(\alpha_p \rho_p u_p)}{\partial t} + \frac{\partial(\alpha_p \rho_p u_p u_p + \alpha_p p_p)}{\partial x} &= -p^I \frac{\partial(\alpha_g)}{\partial x} - \lambda(u_p - u_g) \\
\frac{\partial(\alpha_p \rho_p E_p)}{\partial t} + \frac{\partial(\alpha_p \rho_p E_p u_p + \alpha_p p_p u_p)}{\partial x} &= -p^I u^I \frac{\partial(\alpha_g)}{\partial x} - \mu p^I (p_p - p_g) - \lambda u^I (u_p - u_g)
\end{aligned}$$

where α is the phase volume fraction and the subscripts g and p indicate the flow variable is associated with the gas or particle phase respectively. These equations are determined by taking a phase average over the mixture, see [18, 19, 21, 47] for discussions. The phase flow variables are defined as if they occupied their own spatial locations, i.e. the gas-phase density (ρ_g) is defined as the mass of the gas divided by the volume occupied by the gas. The above system is augmented with the saturation constraint

$$\alpha_g + \alpha_p = 1$$

which relates the volume fractions of the gas and particle phases.

The volume-fraction gradient terms on the right hand side prevent the system from being expressed in conservative form. Thus, they are referred to as non-conservative terms. Their effects are analogous to the volume change in a nozzle, hence they are sometimes called *nozzling* terms [15, 48].

The interface pressure, p^I , and the interface velocity, u^I , are determined through closure relations. There is some debate about the form for these quantities and a number of options have been presented in the literature. Baer and Nunziato [18] developed a model based on continuum mixture theory. This model assumes

$$\begin{aligned} p^I &= p_g \\ u^I &= u_p \end{aligned}$$

where the interface pressure is taken as the gas pressure and the interface velocity is taken as the particle velocity. Saurel and Abgrall [19] used an ensemble averaging procedure to construct a slightly modified model. This model uses the relations

$$\begin{aligned} p^I &= \alpha_g p_g + \alpha_p p_p \\ u^I &= \frac{\alpha_g \rho_g u_g + \alpha_p \rho_p u_p}{\alpha_g \rho_g + \alpha_p \rho_p} \end{aligned}$$

where the interface pressure is defined as the mixture pressure and the interface velocity is the velocity of the center of mass of the bulk material. Additional models are also found in the literature. The interface closure model plays an important role in the behavior of the model and its wave structure.

Mechanical interaction between phases is accounted for. The relaxation parameter μ determines the rate at which pressure equilibrium is achieved. In many multiphase-flow problems, pressure equilibrium occurs very rapidly compared with other processes. Therefore, equilibrium is often assumed to be instantaneous and μ is taken as infinite.

The parameter λ determines the rate at which velocity equilibrium occurs. These terms are used to model the drag force between phases

$$\vec{F} = \lambda(u_p - u_g)$$

accounting for pressure and viscous forces at the phase boundaries. The parameter λ is a function of the particle geometry and the surrounding fluid properties.

Heat transfer between the phases is another important process in multiphase flow. This energy exchange due to conduction and convection can also be accounted for, but the corresponding terms have not been included in the above system.

Although the *seven-equation model* (2.5) has had success in modeling multiphase phenomena, there are some drawbacks. As previously mentioned, the interfacial closure model is not well defined. In addition, numerical solution of the above system poses some challenges. The model is an unconditionally hyperbolic system, which naturally lends itself toward Godunov-type solution methods. However, the associated Riemann problem depends on the entire seven-equation system of coupled non-conservative equations. Although recent progress has been made toward numerically consistent models [48, 49], the coupled system is much more complicated than the corresponding single-phase system.

2.1.3 DEM Method

The objective of the current research is to obtain a set of conservation laws for a mixture of multiple materials. It is desired to avoid some of the drawbacks associated with the seven-equation model discussed in the previous section. This is accomplished using the *discrete equation method* (DEM). The DEM is a powerful methodology which allows for the computation of multiphase fluid flows. The drawbacks of other models are avoided as the interface closure terms are inherent in the model and only single-phase Riemann solutions are computed. This method is an extension to the averaging methods described in Drew and Passman [21] and was originally proposed by Abgrall and Saurel [20].

The averaging method begins by introducing the concept of an indicator function, Ψ_π , defined by:

$$\Psi_\pi(\vec{x}, t) = \begin{cases} 1 & \text{if location } \vec{x} \text{ is occupied by phase } \pi \text{ at time } t, \\ 0 & \text{otherwise.} \end{cases}$$

The indicator is not differentiable at the interface and is therefore not a function in the classical sense. Rather, it is a generalized function [21] and the subsequent derivations must be interpreted with that in mind. The indicator function obeys the evolution equation:

$$\left. \frac{\partial \Psi_\pi(\vec{x}, t)}{\partial t} \right|_{\vec{x}} + u_i^I \left. \frac{\partial \Psi_\pi(\vec{x}, t)}{\partial x_i} \right|_t = 0 \quad (2.6)$$

where u_i^I is the i component of the local interface velocity. This equation states that the material derivative of Ψ_π following the material interface between phases is zero. For points not on the interface, Ψ_π is a constant (either 0 or 1) and therefore the partial derivatives are zero. For points on the interface, Ψ_π is a jump that remains constant and moves with the local interface velocity. Hence, the material derivative is zero at the interface as well.

Following the steps of Chinnayya et al. [22], combine the indicator function with the state and flux tensors using the product rule of differentiation:

$$\begin{aligned} \frac{\partial (\Psi_\pi \mathbf{W})}{\partial t} &= \Psi_\pi \frac{\partial \mathbf{W}}{\partial t} + \mathbf{W} \frac{\partial \Psi_\pi}{\partial t} \\ \frac{\partial (\Psi_\pi \mathbf{H}_i)}{\partial x_i} &= \Psi_\pi \frac{\partial \mathbf{H}_i}{\partial x_i} + \mathbf{H}_i \frac{\partial \Psi_\pi}{\partial x_i} \end{aligned}$$

Adding the above equations together, recalling the transport equation (2.4), and performing some algebraic manipulation, results in the following:

$$\begin{aligned} \frac{\partial (\Psi_\pi \mathbf{W})}{\partial t} + \frac{\partial (\Psi_\pi \mathbf{H}_i)}{\partial x_i} &= \Psi_\pi \left[\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{H}_i}{\partial x_i} \right] + \mathbf{W} \frac{\partial \Psi_\pi}{\partial t} + \mathbf{H}_i \frac{\partial \Psi_\pi}{\partial x_i} \\ &= \Psi_\pi \mathbf{S} + \mathbf{W} \frac{\partial \Psi_\pi}{\partial t} + \mathbf{H}_i \frac{\partial \Psi_\pi}{\partial x_i} \\ &= \Psi_\pi \mathbf{S} - u_i^I \mathbf{W} \frac{\partial \Psi_\pi}{\partial x_i} + \mathbf{H}_i \frac{\partial \Psi_\pi}{\partial x_i} \\ &= \Psi_\pi \mathbf{S} + (\mathbf{H}_i - u_i^I \mathbf{W}) \frac{\partial \Psi_\pi}{\partial x_i} \end{aligned}$$

Thus, the transport of local pure fluids is obtained,

$$\left. \frac{\partial (\Psi_\pi \mathbf{W})}{\partial t} \right|_{\vec{x}} + \left. \frac{\partial (\Psi_\pi \mathbf{H}_i)}{\partial x_i} \right|_t = \Psi_\pi \mathbf{S} + \mathbf{H}_i^{lag} \left. \frac{\partial \Psi_\pi}{\partial x_i} \right|_t \quad (2.7)$$

with

$$\begin{aligned} \mathbf{H}_i^{lag} &\equiv \mathbf{H}_i - u_i^I \mathbf{W} \\ &= (u_i - u_i^I) \mathbf{W} + \mathbf{P}_i \end{aligned} \quad (2.8)$$

defined as the Lagrangian flux. At phase boundaries the Lagrangian flux reduces to

$$\mathbf{H}_i^{lag} = \mathbf{P}_i$$

since the interface velocity is equal to the phase velocity at the interface. Since the gradient of the indicator function is zero everywhere except at phase boundaries, the Lagrangian flux is only evaluated at the interfaces.

Due to the discontinuous nature of Ψ_π , this equation is difficult to deal with in continuous form. Therefore, the numerical method will utilize a weak implementation. Hence, convert this to an integral equation by integrating over the volume of a computational cell, Ω ,

$$\int_{\Omega} \left[\frac{\partial (\Psi_\pi \mathbf{W})}{\partial t} \Big|_{\vec{x}} + \frac{\partial (\Psi_\pi \mathbf{H}_i)}{\partial x_i} \Big|_t \right] d\mathcal{V} = \int_{\Omega} [\Psi_\pi \mathbf{S}] d\mathcal{V} + \int_{\Omega} \left[\mathbf{H}_i^{lag} \frac{\partial \Psi_\pi}{\partial x_i} \Big|_t \right] d\mathcal{V} \quad (2.9)$$

which can be decomposed into separate integrals for the temporal, convective, and non-conservative terms.

2.1.4 ALE Equations

The goal now is to solve the DEM equations on a moving mesh. This is often referred to as an arbitrary Lagrangian Eulerian (ALE) scheme. Saurel et al. [36] used the DEM method on moving meshes for one-dimensional rectangular (Cartesian) geometries. This section will extend that method to higher dimensions and arbitrarily shaped control volume geometries.

We will begin by reproducing (2.9), accounting for the motion of the control

volume

$$\int_{\Omega(t)} \left[\frac{\partial (\Psi_\pi \mathbf{W})}{\partial t} \Big|_{\vec{x}} + \frac{\partial (\Psi_\pi \mathbf{H}_i)}{\partial x_i} \Big|_t \right] d\mathcal{V} = \int_{\Omega(t)} [\Psi_\pi \mathbf{S}] d\mathcal{V} + \int_{\Omega(t)} \left[\mathbf{H}_i^{lag} \frac{\partial \Psi_\pi}{\partial x_i} \Big|_t \right] d\mathcal{V} \quad (2.10)$$

Notice that the integrals are now taken over a moving volume, $\Omega(t)$.

The time-derivative term

$$\int_{\Omega(t)} \frac{\partial (\Psi_\pi \mathbf{W})}{\partial t} \Big|_{\vec{x}} d\mathcal{V}$$

will be addressed first. The first step in a standard finite-volume scheme is to pull the time derivative in (2.10) outside the integral. However, the current term is integrated over a moving control volume, and hence the integration volume is time-dependent. Therefore, as it currently stands, the time derivative can't be pulled outside the integral. This is due to the fixed reference frame in which the equations are expressed. As an alternative, the equation can be expressed in a new reference frame, $\vec{\xi}(\vec{x}, t) \in \mathbb{R}^3$, that moves with the control volume. As seen from this new reference frame, the integration domain does not vary with time. Therefore, through this change of variables, an integral can be transformed, such that

$$\int_{\Omega(t)} (\bullet) d\mathcal{V} \quad \Rightarrow \quad \int_{\Omega_\xi} (\bullet) \mathcal{J} d\mathcal{V}_\xi$$

where $d\mathcal{V} = dx_1 dx_2 dx_3$ is the differential volume in the Eulerian frame and $d\mathcal{V}_\xi = d\xi_1 d\xi_2 d\xi_3$ is the differential volume in the ALE frame. The Jacobian, \mathcal{J} , is defined as the determinant of the Jacobi Matrix, \mathbf{J} :

$$\mathbf{J} = \frac{\partial \vec{x}}{\partial \vec{\xi}} \Big|_t, \quad J_{ij} = \frac{\partial x_i}{\partial \xi_j} \Big|_t$$

$$\mathcal{J} = \det(\mathbf{J}) = \frac{d\mathcal{V}}{d\mathcal{V}_\xi}$$

The chain rule provides expressions for the time derivative in the ALE reference

frame:

$$\begin{aligned}\frac{\partial}{\partial t}\Big|_{\vec{\xi}} &= \frac{\partial}{\partial t}\Big|_{\vec{x}} + \frac{\partial x_i}{\partial t}\Big|_{\vec{\xi}} \frac{\partial}{\partial x_i}\Big|_t \\ &= \frac{\partial}{\partial t}\Big|_{\vec{x}} + \hat{u}_i \frac{\partial}{\partial x_i}\Big|_t\end{aligned}\quad (2.11)$$

where $\hat{u}_i(\vec{x}, t) = \frac{\partial x_i}{\partial t}\Big|_{\vec{\xi}}$ is the velocity of the moving control volume with respect to the fixed reference frame. Note that the form of (2.11) is similar to the definition of the substantial (or material) derivative commonly used in fluid mechanics.

Recalling that the time derivative of the Jacobian is given by the expression:

$$\frac{\partial \mathcal{J}}{\partial t}\Big|_{\vec{\xi}} = \mathcal{J} \frac{\partial \hat{u}_i}{\partial x_i}\Big|_t \quad (2.12)$$

and applying (2.11) to the time derivative in (2.10) gives

$$\begin{aligned}\frac{\partial(\Psi_\pi \mathbf{W})}{\partial t}\Big|_{\vec{x}} &= \frac{\partial(\Psi_\pi \mathbf{W})}{\partial t}\Big|_{\vec{\xi}} - \hat{u}_i \frac{\partial(\Psi_\pi \mathbf{W})}{\partial x_i}\Big|_t \\ &= \frac{1}{\mathcal{J}} \frac{\partial(\mathcal{J} \Psi_\pi \mathbf{W})}{\partial t}\Big|_{\vec{\xi}} - \frac{\Psi_\pi \mathbf{W}}{\mathcal{J}} \frac{\partial \mathcal{J}}{\partial t}\Big|_{\vec{\xi}} - \hat{u}_i \frac{\partial(\Psi_\pi \mathbf{W})}{\partial x_i}\Big|_t \\ &= \frac{1}{\mathcal{J}} \frac{\partial(\mathcal{J} \Psi_\pi \mathbf{W})}{\partial t}\Big|_{\vec{\xi}} - \Psi_\pi \mathbf{W} \frac{\partial \hat{u}_i}{\partial x_i}\Big|_t - \hat{u}_i \frac{\partial(\Psi_\pi \mathbf{W})}{\partial x_i}\Big|_t \\ &= \frac{1}{\mathcal{J}} \frac{\partial(\mathcal{J} \Psi_\pi \mathbf{W})}{\partial t}\Big|_{\vec{\xi}} - \frac{\partial(\hat{u}_i \Psi_\pi \mathbf{W})}{\partial x_i}\Big|_t\end{aligned}$$

The time derivative on the right hand side is now taken with respect to the moving reference frame, which is what was originally desired. In addition, there is an additional convective derivative associated with the motion of the reference frame.

Substituting the right hand side into (2.10) yields:

$$\begin{aligned}\int_{\Omega(t)} \left[\frac{1}{\mathcal{J}} \frac{\partial(\mathcal{J} \Psi_\pi \mathbf{W})}{\partial t}\Big|_{\vec{\xi}} - \frac{\partial(\hat{u}_i \Psi_\pi \mathbf{W})}{\partial x_i}\Big|_t + \frac{\partial(\Psi_\pi \mathbf{H}_i)}{\partial x_i}\Big|_t \right] d\mathcal{V} = \\ \int_{\Omega(t)} [\Psi_\pi \mathbf{S}] d\mathcal{V} + \int_{\Omega(t)} \left[\mathbf{H}_i^{lag} \frac{\partial \Psi_\pi}{\partial x_i}\Big|_t \right] d\mathcal{V}\end{aligned}\quad (2.13)$$

This can be simplified further to get

$$\begin{aligned}\int_{\Omega(t)} \frac{1}{\mathcal{J}} \frac{\partial(\mathcal{J} \Psi_\pi \mathbf{W})}{\partial t}\Big|_{\vec{\xi}} d\mathcal{V} + \int_{\Omega(t)} \frac{\partial(\Psi_\pi \mathbf{H}_i^c)}{\partial x_i}\Big|_t d\mathcal{V} = \\ \int_{\Omega(t)} [\Psi_\pi \mathbf{S}] d\mathcal{V} + \int_{\Omega(t)} \left[\mathbf{H}_i^{lag} \frac{\partial \Psi_\pi}{\partial x_i}\Big|_t \right] d\mathcal{V}\end{aligned}\quad (2.14)$$

where the convective-flux terms have been combined using

$$\begin{aligned} \mathbf{H}_i^c &\equiv \mathbf{H}_i - \hat{u}_i \mathbf{W} \\ &= (u_i - \hat{u}_i) \mathbf{W} + \mathbf{P}_i \end{aligned}$$

The final step is to perform the change of variables operation on the time-derivative term from the fixed to the moving coordinates:

$$\begin{aligned} \int_{\Omega_\xi} \frac{\partial (\mathcal{J} \Psi_\pi \mathbf{W})}{\partial t} \Big|_{\vec{\xi}} d\mathcal{V}_\xi + \int_{\Omega(t)} \frac{\partial (\Psi_\pi \mathbf{H}_i^c)}{\partial x_i} \Big|_t d\mathcal{V} = \\ \int_{\Omega(t)} [\Psi_\pi \mathbf{S}] d\mathcal{V} + \int_{\Omega(t)} \left[\mathbf{H}_i^{lag} \frac{\partial \Psi_\pi}{\partial x_i} \Big|_t \right] d\mathcal{V} \end{aligned} \quad (2.15)$$

In the form above, the volume integral for the time-derivative term is no longer time-dependent. Therefore, the derivative can now be pulled outside the integral, as desired.

2.1.5 Axisymmetric Formulation

The preceding sections derived the conservation equations using a three-dimensional Cartesian coordinate system. Many geometries of interest can be simplified using an axisymmetric invariance assumption. This section will define this approximation and discuss its implementation in the current method.

The axisymmetric approximation begins with a mapping to a cylindrical polar coordinate system. Any point in the Cartesian frame, \vec{x} , can be denoted by the associated cylindrical coordinates, $\vec{\check{x}}$. Variables related to polar coordinates are differentiated from the standard Cartesian variables by the $(\check{\cdot})$ notation. Here, \check{x}_1 is the axial coordinate, \check{x}_2 is the radial coordinate, and \check{x}_3 is the angular coordinate. The cylindrical coordinates are related to the Cartesian coordinates by the equations (see [50])

$$\begin{aligned} x_1 &= \check{x}_2 \cos \check{x}_3 \\ x_2 &= \check{x}_2 \sin \check{x}_3 \\ x_3 &= \check{x}_1 \end{aligned}$$

and inversely

$$\begin{aligned}\check{x}_1 &= x_3 \\ \check{x}_2 &= \sqrt{x_1^2 + x_2^2} \\ \check{x}_3 &= \tan^{-1}\left(\frac{x_2}{x_1}\right)\end{aligned}$$

The axisymmetric assumption reduces the dimensionality of the system by assuming that there is symmetry in the angular direction. Therefore, there is no flow velocity and all the derivatives are zero in the angular (\check{x}_3) direction. Note, that due to the restriction on angular velocity, swirling flows are not considered here. This results in a mapping to a lower dimensional space

$$[x_1, x_2, x_3]^T \in \mathbb{R}^3 \rightarrow [\check{x}_1, \check{x}_2]^T \in \mathbb{R} \times \mathbb{R}^+ = \{\check{x}_1 \in \mathbb{R}, \check{x}_2 \geq 0\}$$

The axisymmetric approximation and its application to conservation laws is discussed in Guardone and Vigevano [51]. The axisymmetric form of the conservation laws, (2.1)–(2.3), is given by

$$\frac{\partial(\check{x}_2 \rho)}{\partial t} + \frac{\partial(\check{x}_2(\rho \check{u}_j))}{\partial \check{x}_j} = 0 \quad (2.16)$$

$$\frac{\partial(\check{x}_2 \rho \check{u}_i)}{\partial t} + \frac{\partial(\check{x}_2(\rho \check{u}_i \check{u}_j - \check{\sigma}_{ji}))}{\partial \check{x}_j} = -\delta_{i2} \check{\sigma}_{33} \quad (2.17)$$

$$\frac{\partial(\check{x}_2 \rho E)}{\partial t} + \frac{\partial(\check{x}_2(\rho \check{u}_j E - \check{\sigma}_{ji} \check{u}_i))}{\partial \check{x}_j} = 0 \quad (2.18)$$

where the velocity consists of an axial and radial component

$$\vec{\check{u}} = \begin{pmatrix} \check{u}_1 \\ \check{u}_2 \end{pmatrix}$$

and the stress tensor is

$$\check{\sigma} = \begin{bmatrix} \check{\sigma}_{11} & \check{\sigma}_{12} & 0 \\ \check{\sigma}_{21} & \check{\sigma}_{22} & 0 \\ 0 & 0 & \check{\sigma}_{33} \end{bmatrix}$$

and must maintain its rank of 3. This is to properly account for the ‘hoop stress’, which is the non-zero normal stress in the out of plane (angular) direction, and factors into the conservation of radial momentum (2.17).

As was done with the Cartesian form of the conservation laws in §2.1.1, the axisymmetric form of the laws can be represented as a tensor equation

$$\frac{\partial \check{\mathbf{W}}}{\partial t} + \frac{\partial \check{\mathbf{H}}_i}{\partial \check{x}_i} = \check{\mathbf{S}} \quad (2.19)$$

where

$$\check{\mathbf{W}}(\vec{\check{x}}, t) = \check{x}_2 \begin{bmatrix} 1/\check{x}_2 \\ \rho \\ \rho \check{u}_1 \\ \rho \check{u}_2 \\ \rho E \end{bmatrix}$$

$$\check{\mathbf{H}}(\vec{\check{x}}, t) = \check{x}_2 \begin{bmatrix} 0 & \rho \check{u}_1 & \rho \check{u}_1 \check{u}_1 - \check{\sigma}_{11} & \rho \check{u}_1 \check{u}_2 - \check{\sigma}_{21} & \rho \check{u}_1 E - \check{\sigma}_{1j} \check{u}_j \\ 0 & \rho \check{u}_2 & \rho \check{u}_2 \check{u}_1 - \check{\sigma}_{12} & \rho \check{u}_2 \check{u}_2 - \check{\sigma}_{22} & \rho \check{u}_2 E - \check{\sigma}_{2j} \check{u}_j \end{bmatrix}$$

$$\check{\mathbf{P}}(\vec{\check{x}}, t) = \check{x}_2 \begin{bmatrix} -\check{u}_1/\check{x}_2 & 0 & -\check{\sigma}_{11} & -\check{\sigma}_{21} & -\sigma_{1j} \check{u}_j \\ -\check{u}_2/\check{x}_2 & 0 & -\check{\sigma}_{12} & -\check{\sigma}_{22} & -\sigma_{2j} \check{u}_j \end{bmatrix}$$

and

$$\check{\mathbf{S}}(\vec{\check{x}}, t) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\check{\sigma}_{33} \\ 0 \end{bmatrix}$$

The divergence term in (2.19) is the standard two-dimensional Cartesian divergence operator in the axial plane. This is an important point, as it means that the above system can be solve using the same methods employed to solve the standard planar equations, without the need for modification to the divergence operation.

The DEM method can then be applied to the axisymmetric conservation laws. Similar to the derivation in §2.1.3, (2.19) can be combined with the transport of an indicator function,

$$\frac{\partial \Psi_\pi}{\partial t} + \check{u}_i^I \frac{\partial \Psi_\pi}{\partial \check{x}_i} = 0 \quad (2.20)$$

to obtain the governing equation for our material

$$\int_{\Omega_\xi} \frac{\partial (\mathcal{J}\Psi_\pi \check{\mathbf{W}})}{\partial t} \Big|_{\check{\xi}} d\mathcal{V}_\xi + \int_{\Omega(t)} \frac{\partial (\Psi_\pi \check{\mathbf{H}}_i^c)}{\partial \check{x}_i} \Big|_t d\mathcal{V} = \int_{\Omega(t)} \Psi_\pi \check{\mathbf{S}} d\mathcal{V} + \int_{\Omega(t)} \check{\mathbf{H}}_i^{lag} \frac{\partial \Psi_\pi}{\partial \check{x}_i} \Big|_t d\mathcal{V} \quad (2.21)$$

This equation has identical form as (2.15). The difference is the modification of the state vector and flux to include a dependence on radial position and the inclusion of the additional source term. In practice the radius term is incorporated into the volume and face area calculations.

2.1.6 Spatial Discretization

The discretization technique used here is based on the finite-volume method. In this method, the computational domain is split into finite cells (Figure 2.1). The governing equations are then solved over each computational cell using approximations to the flow-field. General concepts associated with the finite-volume method can be found in a number of texts, such as Hirsch [52] or Toro [53].

The current method considers each computational cell as a representative multiphase control volume. Thus, each computational cell is further subdivided into separate volumes occupied by each material constituent. These regions are defined by the indicator function, Ψ_π . Averages are taken over each cell, obtaining algebraic relations for the flow variables. The averages are found by examining all the contact surfaces within the cell and summing all of the interface contributions.

Notice that there is a fundamental difference between the current discretization and the seven-equation model discussed in §2.1.2. The seven-equation model treats each phase continuously. Thus, it is assumed both gas and particle material are co-located at every point in space. A smoothly varying volume fraction determines how much of each material is actually at each point. Therefore, there is no concept of material interfaces within the computational cell or at the cell boundaries. Within the DEM framework, each material is assumed to occupy its own space. The material interfaces between phases are persistent. Another way to look at this

is that the seven-equation model does its averaging at the global equation level, while the DEM averages the mixture at the discrete level.

2.1.7 Numerical Integration of Terms

Previously, the conservation laws for the pure components of a multiphase mixture were derived for a moving control volume, resulting in (2.15). This section will proceed to develop the discrete form of the equation and discuss how to evaluate each term. The numerical discretization follows Chinnayya et al. [22], while extending the method to account for the moving control volume. Whereas the original reference relied on a structured Cartesian mesh, the current description is presented for arbitrary spatial decomposition, although in practice only 8-node 3D brick elements and 4-node 2D quadrilateral elements have been utilized.

First, decompose (2.15) such that

$$I_T + I_C = I_S + I_L \quad (2.22)$$

with

$$I_T = \int_{\Omega_\xi} \left. \frac{\partial (\mathcal{J} \Psi_\pi \mathbf{W})}{\partial t} \right|_{\bar{\xi}} d\mathcal{V}_\xi \quad (2.23)$$

$$I_C = \int_{\Omega(t)} \left. \frac{\partial (\Psi_\pi \mathbf{H}_i^c)}{\partial x_i} \right|_t d\mathcal{V} \quad (2.24)$$

$$I_S = \int_{\Omega(t)} \Psi_\pi \mathbf{S} d\mathcal{V} \quad (2.25)$$

$$I_L = \int_{\Omega(t)} \mathbf{H}_i^{lag} \left. \frac{\partial \Psi_\pi}{\partial x_i} \right|_t d\mathcal{V} \quad (2.26)$$

The above integrals will be computed numerically to form the discrete equations for each phase.

As macroscopic representations of sub-scale phenomena will be formed, a number of averaging procedures can be defined. Throughout this document, the notation of an over-bar ($\bar{\cdot}$) will be used to indicate an averaged quantity. Rather than complicate the nomenclature with distinct notation for each averaging operation, this operator will be overloaded and used for many types of averages.

The desired operation should be apparent within the context of its use. Some commonly used averaging procedures are

Volume Average

$$\bar{f} = \frac{1}{\Omega} \int_{\Omega(t)} f d\mathcal{V}$$

Surface Average

$$\bar{f} = \frac{1}{S} \oint_{\partial\Omega_f(t)} f d\mathcal{A}$$

Phase-Volume Average

$$\begin{aligned} \bar{f}_\pi &= \frac{1}{\Omega_\pi} \int_{\Omega(t)} \Psi_\pi f d\mathcal{V} \\ &= \frac{1}{\Omega_\pi} \int_{\Omega_\pi(t)} f d\mathcal{V} \end{aligned}$$

The above averages use the cell volume

$$\Omega \equiv \int_{\Omega(t)} d\mathcal{V}$$

as well as the volume occupied by phase π within a grid cell, which is given as

$$\Omega_\pi \equiv \int_{\Omega(t)} \Psi_\pi d\mathcal{V}$$

The volume fraction of phase π in a cell is defined as

$$\alpha_\pi = \frac{\Omega_\pi}{\Omega}$$

and the mass fraction is defined as

$$x_\pi = \frac{\int_{\Omega(t)} \rho \Psi_\pi d\mathcal{V}}{\int_{\Omega(t)} \rho d\mathcal{V}}$$

where ρ is the material density.

2.1.7.1 Time Derivative Term

The first integral, I_T , reads

$$I_T = \int_{\Omega_\xi} \left. \frac{\partial (\mathcal{J} \Psi_\pi \mathbf{W})}{\partial t} \right|_{\bar{\xi}} d\mathcal{V}_\xi \quad (2.27)$$

Recalling that Ω_ξ is invariant with time, the time derivative may be pulled out of the integral and simplified as

$$I_T = \frac{\partial}{\partial t} \left[\int_{\Omega_\xi} \Psi_\pi \mathbf{W} \mathcal{J} d\mathcal{V}_\xi \right]_{\vec{\xi}} \quad (2.28)$$

$$= \frac{\partial}{\partial t} \left[\int_{\Omega(t)} \Psi_\pi \mathbf{W} d\mathcal{V} \right]_{\vec{\xi}} \quad (2.29)$$

$$= \frac{\partial}{\partial t} (\Omega_\pi \overline{\mathbf{W}}_\pi) \Big|_{\vec{\xi}} \quad (2.30)$$

$$= \frac{\partial}{\partial t} (\Omega \alpha_\pi \overline{\mathbf{W}}_\pi) \Big|_{\vec{\xi}} \quad (2.31)$$

where the over-bar indicates the phase-volume averaging operator. Further discretization of this term depends on the time-stepping scheme used and will be discussed in a later section.

2.1.7.2 Convective Term

The convective integral can be simplified by applying the divergence theorem

$$I_C = \int_{\Omega(t)} \frac{\partial (\Psi_\pi \mathbf{H}_i^c)}{\partial x_i} \Big|_t d\mathcal{V} \quad (2.32)$$

$$= \oint_{\partial\Omega(t)} \Psi_\pi \mathbf{H}_i^c n_i d\mathcal{A} \quad (2.33)$$

$$= \oint_{\partial\Omega(t)} \Psi_\pi \mathbf{F}^c d\mathcal{A} \quad (2.34)$$

The outward normal is denoted \vec{n} , and $\mathbf{F}^c \equiv \mathbf{H}_i^c n_i$ is the outward flux normal to the surface. As the surface of the control volume is composed of a number of distinct faces, the surface integral can be split into components over each face and a surface average can be taken over the face

$$I_C = \sum_f \oint_{\partial\Omega_f(t)} \Psi_\pi \mathbf{F}^c d\mathcal{A} \quad (2.35)$$

$$= \sum_f [S \overline{\Psi_\pi \mathbf{F}^c}]_f \quad (2.36)$$

This can be further decomposed into phase contacts at the face surface

$$I_C = \sum_f \left[\sum_{pq} (S \Psi_\pi^* \mathbf{F}^{c*})_{pq} \right]_f \quad (2.37)$$

where the subscripts indicate the contact surface between phase p and phase q across face f . These phase contacts are defined as the surface of the material interface between phases (See [22] for details). The contact surfaces are computed from Table 2.1 and the indicator function is evaluated at the contact using Table 2.2.

2.1.7.3 Source Term

The source term is handled by a simple integration over the cell.

$$I_S = \int_{\Omega(t)} \Psi_\pi \mathbf{S} d\mathcal{V} \quad (2.38)$$

$$= \Omega_\pi \bar{\mathbf{S}}_\pi \quad (2.39)$$

$$= \Omega \alpha_\pi \bar{\mathbf{S}}_\pi \quad (2.40)$$

Thus, single-point Gauss quadrature is used to evaluate the source term, where the average value of the source, $\bar{\mathbf{S}}_\pi$, is evaluated at the cell centroid.

2.1.7.4 Lagrangian/Non-conservative Source Term

The discretization of the Lagrangian source term will be addressed next,

$$I_L = \int_{\Omega(t)} \mathbf{H}_i^{lag} \left. \frac{\partial \Psi_\pi}{\partial x_i} \right|_t d\mathcal{V} \quad (2.41)$$

Noting that Ψ_π is constant over the phase, $\partial \Psi_\pi / \partial x$ (and thus the above integral) is only non-zero at the material or phase boundaries. In addition, \mathbf{H}^{lag} is uniform within the phase. Therefore, if Σ is the surface defined by the union of all interfaces between phases then the volume integral can be replaced by a surface integral over the interface

$$I_L = \oint_{\Sigma(t)} [[\Psi_\pi]] \mathbf{H}_i^{lag} n_i d\mathcal{A} \quad (2.42)$$

$$= \oint_{\Sigma(t)} [[\Psi_\pi]] \mathbf{F}^{lag} d\mathcal{A} \quad (2.43)$$

where $[[\Psi_\pi]]$ is the jump in the normal direction across the interface, and the normal flux is given by $\mathbf{F}^{lag} \equiv \mathbf{H}_i^{lag} n_i$. The unit normal in these equations is defined pointing outward from phase π into the other phase.

Note that this term is very different from the convective-flux term, I_C . In the convective term, the divergence theorem was applied resulting in a conservative flux which was passed across the cell boundaries from one cell to another. In the evaluation of the Lagrangian term, I_L , the divergence theorem has not been employed, rather the integral reduces to a surface integral over all contacts within the cell and on its boundary. This results in a non-conservative flux which does not pass between cells, but rather passes from one phase to another within the same cell. Thus from an algorithmic standpoint, the convective flux is face-based, while the Lagrangian flux is cell-based.

To evaluate the above term, the surface integral is broken up into boundary, internal, and numerical components.

$$\begin{aligned}
I_L &= \oint_{\Sigma^{boundary}(t)} [[\Psi_\pi]] \mathbf{F}^{lag} d\mathcal{A} + \\
&\quad \oint_{\Sigma^{internal}(t)} [[\Psi_\pi]] \mathbf{F}^{lag} d\mathcal{A} + \\
&\quad \oint_{\Sigma^{numerical}(t)} [[\Psi_\pi]] \mathbf{F}^{lag} d\mathcal{A} \tag{2.44}
\end{aligned}$$

$$= I_L^{boundary} + I_L^{internal} + I_L^{numerical} \tag{2.45}$$

The boundary terms result from material interfaces located at the cell boundaries which arise due to variation in volume fraction between cells. Thus, this term can be decomposed into integrals over each face of the cell, and then further decomposed by each individual phase contact surface.

$$I_L^{boundary} = \oint_{\Sigma^{boundary}(t)} [[\Psi_\pi]] \mathbf{F}^{lag} d\mathcal{A} \tag{2.46}$$

$$= \oint_{\partial\Omega(t)} [[\Psi_\pi]] \mathbf{F}^{lag} d\mathcal{A} \tag{2.47}$$

$$= \sum_f \oint_{\partial\Omega_f(t)} [[\Psi_\pi]] \mathbf{F}^{lag} d\mathcal{A} \tag{2.48}$$

$$= \sum_f \left[S [[\Psi_\pi]] \mathbf{F}^{lag} \right]_f \tag{2.49}$$

$$= \sum_f \left[\sum_{pq} (S [[\Psi_\pi]]^* \mathbf{F}^{lag*})_{pq} \right]_f \tag{2.50}$$

where the jumps in the indicator function are evaluated at the contact using Table 2.3.

The internal terms result from material interfaces within the cell. As these interfaces are not adequately resolved at the scale of the mesh, they require some kind of sub-grid model to define them. Calculation of these terms directly is discussed in Chinnayya et al. [22]. However, in practice it is found that these terms act as a type of relaxation process, and that is how the current method evaluates them.

$$I_L^{internal} \rightarrow \textit{relaxation process} \quad (2.51)$$

The relaxation process used to approximate these terms is discussed in more detail in a later section.

Finally, the numerical term arises due to specifics in the numerical scheme. Specifically, it is due to the higher-order extensions to the method, which will be discussed in a later section. For the higher-order methods the material state is not assumed to be constant within the computational cell. Thus, numerical interfaces have been introduced which must be accounted for. This is discussed in Abgrall and Saurel [20] and results in a term which is similar to the boundary term, but evaluated within the cell at the numerical interface.

$$I_L^{numerical} = \left[\sum_{pq} (S [[\Psi_\pi]]^* \mathbf{F}^{lag*})_{pq} \right]_{H.O.} \quad (2.52)$$

2.1.8 Continuous Limit of Discrete Equations

As derived in the previous sections, the DEM provides an elegant framework for analyzing multiphase mixtures. Yet, the governing equations (2.15) only make sense at a discrete level. Often it is instructive to analyze a computational model at the differential level in order to understand the model structure and properties. Chinnayya et al. [22] derived a continuous analogue of the DEM, resulting in a system of partial differential equations. For a two-phase gas-particle mixture these

equations are given in a one-dimensional fixed reference frame by

$$\begin{aligned}
\frac{\partial(\alpha_g)}{\partial t} &= -u^I \frac{\partial(\alpha_g)}{\partial x} - \mu (p_p - p_g) \\
\frac{\partial(\alpha_g \rho_g)}{\partial t} + \frac{\partial(\alpha_g \rho_g u_g)}{\partial x} &= 0 \\
\frac{\partial(\alpha_g \rho_g u_g)}{\partial t} + \frac{\partial(\alpha_g \rho_g u_g u_g + \alpha_g p_g)}{\partial x} &= p^I \frac{\partial(\alpha_g)}{\partial x} + \lambda (u_p - u_g) \\
\frac{\partial(\alpha_g \rho_g E_g)}{\partial t} + \frac{\partial(\alpha_g \rho_g E_g u_g + \alpha_g p_g u_g)}{\partial x} &= p^I u^I \frac{\partial(\alpha_g)}{\partial x} + \mu p^I (p_p - p_g) + \lambda u^I (u_p - u_g) \\
\frac{\partial(\alpha_p \rho_p)}{\partial t} + \frac{\partial(\alpha_p \rho_p u_p)}{\partial x} &= 0 \\
\frac{\partial(\alpha_p \rho_p u_p)}{\partial t} + \frac{\partial(\alpha_p \rho_p u_p u_p + \alpha_p p_p)}{\partial x} &= -p^I \frac{\partial(\alpha_g)}{\partial x} - \lambda (u_p - u_g) \\
\frac{\partial(\alpha_p \rho_p E_p)}{\partial t} + \frac{\partial(\alpha_p \rho_p E_p u_p + \alpha_p p_p u_p)}{\partial x} &= -p^I u^I \frac{\partial(\alpha_g)}{\partial x} - \mu p^I (p_p - p_g) - \lambda u^I (u_p - u_g)
\end{aligned} \tag{2.53}$$

where α is the phase volume fraction and the subscripts g and p indicate the flow variable is associated with the gas or particle phase respectively. The equations were obtained by taking the continuous limit of the discrete form of (2.15) as the time-step and spatial volume tend towards zero.

The constants on the right hand side of (2.53) come directly from applying the limiting process to the non-conservative terms (2.26). The resulting constants are functions of a characteristic surface area, s , and the acoustic impedance of the phase, given by $Z = (\rho c)$ where ρ and c are the density and sound speed of the phase respectively. The pressure and velocity relaxation coefficients are

$$\begin{aligned}
\mu &= \frac{s}{Z_g + Z_p} \\
\lambda &= \frac{s Z_g Z_p}{2(Z_g + Z_p)}
\end{aligned}$$

respectively. The interface pressure and velocity are derived as

$$\begin{aligned}
p^I &= \frac{Z_p p_g + Z_g p_p}{Z_g + Z_p} \pm \frac{Z_g Z_p (u_p - u_g)}{Z_g + Z_p} \\
u^I &= \frac{Z_g u_g + Z_p u_p}{Z_g + Z_p} \pm \frac{(p_p - p_g)}{Z_g + Z_p}
\end{aligned} \tag{2.54}$$

where the sign of the \pm operator is determined from the sign of the volume fraction gradient, $\frac{\partial \alpha_g}{\partial x}$. Notice that the form of the above coefficients are a direct consequence of Chinnayya's choice to use the acoustic Riemann solver approximation (see [53]).

It is observed that the form of (2.53) is identical to the seven-equation model (2.5). The difference lies in the expressions for the constants on the right hand side. For the continuous limit of the DEM, the analytic expressions for the constants are consequences of the limiting process and are therefore implicitly defined by the method. The constants used in the seven-equation model, however, must be determined through a closure model, as discussed in §2.1.2.

It is noticed that for $Z_p \gg Z_g$ and assuming that the differences in the phase velocities and pressures are much smaller than their magnitude, then the interface values (2.54) reduce to the values used in the BN model (2.6). These conditions are reasonably represented within dilute gas-particle flows, but are not valid for detonation waves where the acoustic impedances are of the same order.

The system (2.53) can be expressed in terms of primitive variables

$$\frac{\partial \mathbf{w}_\pi}{\partial t} + \mathbf{a}_\pi \frac{\partial \mathbf{w}_\pi}{\partial x} = \mathbf{s}_\pi \quad (2.55)$$

where the subscript π can be either g or p for the gas or particle phase respectively. The vector of primitive variables is defined as

$$\mathbf{w}_\pi = \begin{bmatrix} \alpha \\ \rho \\ u \\ p \end{bmatrix} \quad (2.56)$$

and \mathbf{s}_π represents the source term. The Jacobian matrix is given by

$$\mathbf{a}_\pi = \begin{bmatrix} u^I & 0 & 0 & 0 \\ \frac{\rho_\pi}{\alpha_\pi}(u_\pi - u^I) & u_\pi & \rho_\pi & 0 \\ \frac{p_\pi - p^I}{\alpha_\pi \rho_\pi} & 0 & u_\pi & \frac{1}{\rho_\pi} \\ \frac{\rho_\pi (c^I)^2}{\alpha_\pi}(u_\pi - u^I) & 0 & \rho_\pi c_\pi^2 & u_\pi \end{bmatrix} \quad (2.57)$$

where c^I is an interface sound speed.

Chinnayya et al. [22] provides detailed mathematical analysis of the continuous limit of the DEM equations. They show that the eigenvalues of the characteristic

matrix, \mathbf{a}_π , are u^I , u_π , $(u_\pi + c_\pi)$, and $(u_\pi - c_\pi)$. Since all eigenvalues are real, the system is always hyperbolic. In addition, it is observed that the system is conservative with respect to the mixture. Chinnayya also shows that the entropies of the phases are always non-decreasing, thus the system satisfies the second law of thermodynamics.

2.1.9 Time-stepping Scheme

Using the discretization techniques discussed in §2.1.7, (2.22) can be rearranged as

$$I_T = -I_C + I_S + I_L^{boundary} + I_L^{numerical} + I_L^{internal} = R(t) \quad (2.58)$$

where $R(t)$ represents the combined right hand side (RHS). After the spatial discretization, R is a function only of time. Therefore, along with (2.31), the system is represented by the ODE

$$\frac{d}{dt} (\Omega \alpha_\pi \overline{\mathbf{W}}_\pi) = R(t) \quad (2.59)$$

which is solved for each grid cell.

As seen in (2.51), the terms associated with internal contacts ($I_L^{internal}$) are not computed directly, but are rather represented using a sub-grid relaxation model. These relaxation processes are handled using a Strang-type operator-split approach [54]. Therefore, the RHS is split

$$R(t) = R^{transport}(t) + R^{relax}(t) \quad (2.60)$$

where $R^{transport}$ and R^{relax} represent the contributions from the hydrodynamic transport and relaxation components respectively.

The hydrodynamic transport terms are computed first. A number of choices are possible for discretizing the temporal derivative. The current method uses a leap-frog type time-stepping scheme

$$\frac{(\Omega \alpha_\pi \overline{\mathbf{W}}_\pi)^{(n+1)*} - (\Omega \alpha_\pi \overline{\mathbf{W}}_\pi)^n}{\Delta t} = (R^{transport})^{n+\frac{1}{2}} \quad (2.61)$$

where the time-step size is given by $\Delta t = t^{n+1} - t^n$. The superscripts in (2.61) indicate that the given quantity is evaluated at the specified time level. The $(n+1)^*$ term designates that the quantities have only been updated to account for the transport terms and must still be updated for the relaxation processes. The RHS is evaluated at the mid-level of the time-step, $t^{n+\frac{1}{2}} = (t^{n+1} + t^n)/2$. Advancing the material state to $t^{n+\frac{1}{2}}$ is discussed later in §2.1.13. The leap-frog scheme is formally second-order accurate for constant time-steps. If the RHS is evaluated at the old time level, t^n , then this scheme reduces to the forward Euler time-stepping scheme and the order of accuracy is reduced to first order.

Once the hydrodynamic transport has been integrated, the state can be updated to account for relaxation processes. Thus,

$$\frac{(\Omega\alpha_\pi \overline{\mathbf{W}}_\pi)^{(n+1)} - (\Omega\alpha_\pi \overline{\mathbf{W}}_\pi)^{(n+1)*}}{\Delta t} = (R^{relax}) \quad (2.62)$$

is solved, where the $(n+1)$ superscript notates the final state at the end of the time-step. The RHS is evaluated using the state at the end of the transport update. The evaluation of the relaxation terms is first-order in time. Details of this step are discussed later in §2.1.14.

2.1.9.1 Time-stepping Stability

The evaluation of the transport equations consists of evaluating Riemann problems between the phase contacts on the cell boundaries. Thus, the time-stepping scheme behaves like a Godunov-type method. Therefore, the stability constraint on the time-step size is given by

$$\frac{\Delta t}{\Delta x} \max_{\pi} (|u_\pi| + c_\pi) \leq 1 \quad (2.63)$$

where the maximum is evaluated over the states of all phases. Here, Δx is taken as a characteristic length of the cell. This constraint ensures that the waves coming from each face do not cross the opposite cell boundaries. A more detailed analysis of this constraint is found in Chinnayya et al. [22].

2.1.10 Geometric Conservation Laws

When constructing a numerical scheme, care must be taken that the geometric quantities are computed consistently with the underlying mathematics. This is especially important for ALE computations using moving and/or deforming grids. To ensure that the conservation property of the equations is not violated, the geometric conservation laws (GCLs) [30–35] must be satisfied along with the governing set of equations. Actually, the GCLs are implicitly coupled to the conservation equations as they can be derived using simplifying assumptions for the flow. The GCLs are composed of two laws dealing with surface and volume effects, which will be discussed separately.

The surface conservation law (SCL) is obtained by assuming a uniform flow on a non-moving mesh. It is expressed in continuous form as

$$\oint_{\partial\Omega(t)} n_i d\mathcal{A} = 0 \quad (2.64)$$

and simply states that the control volume must be surrounded by a closed boundary. This is a fundamental mathematical property that was required for use of the divergence theorem in the derivation of the flux form of the equations. Note that this is not dependent on a moving control volume in any way. Thus, the SCL must be satisfied for computations on static as well as dynamic meshes.

Subdividing the cell boundary into sub-surfaces (or faces), a discrete representation of the SCL is obtained by summing over all faces, f , surrounding the computational cell, such that

$$\sum_f (\bar{S}\bar{n}_i)_f = 0 \quad (2.65)$$

from which the requirements for the definition of the numerical representation of the surface vectors is obtained. It is essential that the SCL is satisfied when defining the discrete representation of the surface areas (\bar{S}) and normal vectors (\bar{n}). This is fairly trivial when dealing with planar face geometries, for example straight lines for 2D elements or triangular faces for 3D tetrahedral elements, because the area

and normal vectors can be computed exactly, such that $n_i(\vec{x}, t) = \bar{n}_i$. However, for non-planar face geometries, like the quadrilateral faces associated with 3D brick elements, this can be a little more complicated since the continuous normal vector varies along the face, such that $n_i(\vec{x}, t) \neq \bar{n}_i$. Thus, the SCL is intimately related to the definition of faces. The reader is referred to Zhang et al. [34] for a more detailed description of the SCL.

The other GCL, referred to as the volume conservation law (VCL), is given by

$$\left. \frac{\partial \Omega^c}{\partial t} \right|_{\vec{\xi}} - \oint_{\partial \Omega(t)} \hat{u}_i n_i d\mathcal{A} = 0 \quad (2.66)$$

which states that as a control volume moves through space its change in volume is equal to the volume swept out by the moving boundary. The discrete representation is

$$\Delta \Omega^{n+1} = \Omega^{n+1} - \Omega^n = \Delta t \sum_f (\overline{S \hat{u}_n})_f^{n+\frac{1}{2}} \quad (2.67)$$

which states that during a time-step, the total increase in volume for a computational cell is equal to the sum of the volume flux across the faces. Evaluation of the right hand side terms must be determined from the underlying time-stepping algorithm for the flow solver. Here, the time level $\{n + \frac{1}{2}\}$ is used in order to be consistent with the leap-frog time-stepping scheme (2.61).

Notice that error has been introduced into the solution at this point. In the integral equation (2.66) the normal mesh velocity, $\hat{u}_i n_i$, is a function of time as well as the spatial location along the boundary. Hence it accounts for the actual variable translation, rotation, and deformation of the surrounding boundary. In the discrete equation (2.67), the normal velocity has been replaced by an average normal mesh velocity, $\overline{\hat{u}_n}$, which is constant over the time-step. This approximation does not account for rotation or deformation of the face. This is fundamentally where the problem lies. The Riemann solution used to approximate the convective flux requires that the reference frame is moving with a single constant velocity. This is because the Riemann problem is a 1D solution of the conservation laws, projected

along the normal vector. However, if the normal vector does not point in a constant direction, i.e. the face rotates, then this is no longer a 1D problem. Rather than trying to find a solution to the Riemann problem in a rotating reference frame, it is assumed that the face translates in a constant direction, and the geometric approximations are chosen to be consistent with (2.67).

One way to satisfy the VCL is to compute the volume, Ω^{n+1} , using (2.67). However, after many time-steps, this computed volume could drift significantly from the actual geometrically computed cell volume based on the cell coordinates. Thus, it is more desirable to use the actual geometric cell volume and modify the parameters on the right hand side of (2.67) to obtain consistency.

Start by looking at an individual face f . Define the face volume flux, $\Delta\Omega_f^{n+1}$, as the geometric volume swept out by the motion of the face between time t^n and time t^{n+1} . A graphical representation of this is shown in Figure 2.2. From purely geometrical arguments, one can show that

$$\Delta\Omega^{n+1} = \sum_f \Delta\Omega_f^{n+1}$$

where the cell volume and the face volume fluxes are computed *exactly* using the definition of the cell geometry. Intuitively, one can combine this definition with (2.67) to obtain

$$\Delta\Omega_f^{n+1} = \Delta t (\overline{S\hat{u}_n})_f^{n+\frac{1}{2}} \quad (2.68)$$

A number of paths can be followed from here. For 1D problems, Smith [30] picked the time-step level for which to evaluate the surface area in order to be consistent. Lesoinne and Farhat [31], subdivided cell faces into triangular sub-faces to be consistent. The approach taken here is to compute the normal mesh velocity based on the geometric volume flux, such that

$$(\hat{u}_n)_f^{gcl} = \frac{\Delta\Omega_f^{n+1}}{\overline{S}_f^{n+\frac{1}{2}} \Delta t} \quad (2.69)$$

Note that the time level has been replaced by the superscript *gcl* to indicate that the normal velocity is not computed at the time level consistent with the other

variables, but rather at a state consistent with the GCL. Computing the normal velocity from (2.69) gives a consistent 1D representation of the mesh motion for the flux calculation. This is the same normal velocity used in Vorobiev et al. [55].

Figure 2.3 shows the importance of obeying the GCLs. The figure shows the computed density of a uniform, zero-velocity flow. Initially, in Figure 2.3(a), the mesh is Cartesian and the density is at a constant value, indicated by the constant green coloring. After the time-step, the mesh has moved to a new location. Figure 2.3(b) shows the results obtained using surface areas and normals naively taken from the beginning of the time-step. Note that the density is no longer uniform throughout the geometry, indicated by the red, green, and blue shading of the cells. Thus the mesh motion results in a non-physical mass flux between cells. Figure 2.3(c) shows the results of the same calculation using a GCL consistent mesh velocity obtained from (2.69). Here the correct, constant density result is computed. Thus, failure to properly account for the GCLs can have large impacts on the quality of the computed solution.

As a side note, issues with the GCLs are generally handled implicitly in Lagrange-Remap ALE schemes. In the direct-ALE scheme discussed here, the Lagrange flux and convective flux are coupled together, thus requiring the same surface area and normals to be used for each. In the remap approach, the advective flux is generally computed directly from the volume flux, thus solving (2.69) automatically. This is possible because the Lagrange and convective fluxes are decoupled allowing the geometric quantities to be computed differently for each operation.

2.1.11 Flux Calculation

The evaluation of the discrete convective (2.37) and Lagrangian (2.50) terms requires the computation of single-phase fluxes at cell boundaries. These fluxes must be evaluated at inter-phase contacts as well as contacts between differing phases. These fluxes are obtained by solving the Riemann problem [53]. The original Godunov [56] method used an exact Riemann solver for gas dynamics

using an ideal-gas equation of state. However, for more complex material equations, exact solvers are often not available. Fortunately, approximate Riemann solvers are available and capable of achieving the desired level of accuracy.

The Riemann problem is an initial value problem consisting of a conservation law together with piecewise-constant initial conditions having a single discontinuity. For the one-dimensional Euler equations, the Riemann problem is given by

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0 \quad (2.70)$$

where \mathbf{W} and \mathbf{F} are the one-dimensional analogues to the state and flux tensors used in (2.4). The initial conditions are

$$\mathbf{W}(x, 0) = \begin{cases} \mathbf{W}_L & \text{if } (x < 0) \\ \mathbf{W}_R & \text{if } (x > 0) \end{cases} \quad (2.71)$$

where the subscripts L and R designate the states on the left and right of the initial discontinuity located at $x = 0$. Solving the Riemann problem is done by a similarity solution

$$\mathbf{W}(x, t) = \mathbf{W}(x/t) \quad (2.72)$$

as depicted in Figure 2.4. The structure of the solution consists of four constant states (\mathbf{W}_L , \mathbf{W}_L^* , \mathbf{W}_R^* , and \mathbf{W}_R) separated by three waves corresponding to the eigenvalues of the PDE ($(u-c)$, u , and $(u+c)$). The waves on the left and right can be either rarefaction waves or shock waves. The center wave is always a contact discontinuity which moves with the material velocity.

The Acoustic Riemann Solver

Obtaining the Riemann solution requires the determination of the *star* state between the waves, \mathbf{W}_L^* and \mathbf{W}_R^* . It is noted that the pressure and velocity on each side of the contact wave are equal to each other ($p_L^* = p_R^*$, $u_L^* = u_R^*$), while the densities are different ($\rho_L^* \neq \rho_R^*$). Therefore, the pressure and velocity in the middle of the waves are represented by the singular values p^* and u^* .

Exact solutions for the star state are only available for simplified equations of state. Therefore, the current method employs a linearized solution commonly referred to as the acoustic Riemann solver [53, 57]. Integrating along the $(u + c)$ and $(u - c)$ characteristics results in

$$p^* = p_L + Z_L(u_{nL} - u^*) \quad (2.73)$$

$$p^* = p_R + Z_R(u^* - u_{nR}) \quad (2.74)$$

where the subscripts L and R indicate that the quantities are obtained from the left and right states as defined by the initial conditions (2.71). The velocity is designated as u_n to indicate that it is taken in the direction normal to the interface. The acoustic impedance of the material is given by $Z = (\rho c)$, where c is the sound speed. Combining (2.73) and (2.74) and solving for u^* gives

$$u^* = \frac{Z_L u_{nL} + Z_R u_{nR} + (p_L - p_R)}{Z_L + Z_R} \quad (2.75)$$

The pressure is found by taking the average of (2.73) and (2.74) and using (2.75)

$$\begin{aligned} p^* &= \frac{1}{2} [p_L + p_R + Z_L(u_{nL} - u^*) + Z_R(u^* - u_{nR})] \\ &= \frac{Z_R p_L + Z_L p_R + Z_L Z_R (u_{nL} - u_{nR})}{Z_L + Z_R} \end{aligned} \quad (2.76)$$

The AUFS Flux

After using the acoustic solution to determine the star state, the full flux vector can be determined. The current methodology requires a convective flux when evaluating (2.37) and a Lagrangian flux when evaluating (2.50). These fluxes are calculated by the Artificially Upstream Flux-vector Splitting (AUFS) scheme of Sun and Takayama [58].

The numerical scheme is implemented using the finite-volume method for a moving unstructured grid system where cell boundaries are not necessarily oriented along coordinate directions. Therefore, the flux must be evaluated in an arbitrary direction. In addition, the mesh is allowed to move. Thus, the flux must be evaluated with respect to a moving frame. Thus, the governing equations (2.4) are

transformed as

$$\left. \frac{\partial \mathbf{W}}{\partial t} \right|_{\bar{\xi}} + \left. \frac{\partial \mathbf{F}^c}{\partial n} \right|_t = 0 \quad (2.77)$$

where $\frac{\partial \mathbf{F}^c}{\partial n}$ is the directional derivative in the direction of the unit normal \vec{n} . As before, the state tensor is given by

$$\mathbf{W} = \begin{bmatrix} 1 \\ \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{bmatrix}$$

and the normal convective flux is

$$\mathbf{F}^c = (u_n - \hat{u}_n) \mathbf{W} + \mathbf{F}^{lag} \quad (2.78)$$

where the Lagrangian flux is

$$\mathbf{F}^{lag} = \begin{bmatrix} -u_n \\ 0 \\ n_1 p \\ n_2 p \\ n_3 p \\ p u_n \end{bmatrix}$$

and the normal velocity is defined by $u_n = n_i u_i$. The mesh velocity, \hat{u}_n , is computed with (2.69). Note that unlike (2.4), the above equations use the pressure rather than the full stress tensor since materials with strength are not considered here. A description of the flux calculation for materials that have strength is found in Knap et al. [59].

The convective flux has been decomposed into a sum of an advective and a Lagrangian component in (2.78). The AUFS technique uses this decomposition

and further breaks up the flux as

$$\mathbf{F}^c = (1 - M)\mathbf{F}_1^c + (M)\mathbf{F}_2^c \quad (2.79)$$

with

$$\begin{aligned} \mathbf{F}_1^c &= (u_n - \hat{u}_n - s_1)\mathbf{W} + \mathbf{F}^{lag} \\ \mathbf{F}_2^c &= (u_n - \hat{u}_n - s_2)\mathbf{W} + \mathbf{F}^{lag} \\ M &= \frac{s_1}{s_1 - s_2} \end{aligned}$$

where s_1 and s_2 are artificial wave speeds. A number of variations on the AUFS solver can be obtained by the choice of these constants. The choice advocated by the original AUFS article was to set s_1 equal to the material velocity in order to resolve contact discontinuities. That choice has been modified here to account for the moving grid, such that

$$\begin{aligned} s_1 &= (u_n - \hat{u}_n) \\ s_2 &= \begin{cases} (u_n - \hat{u}_n) - c & \text{if } (s_1 > 0) \\ (u_n - \hat{u}_n) + c & \text{if } (s_1 \leq 0) \end{cases} \end{aligned}$$

for which \mathbf{F}_1^c becomes equal to the Lagrangian flux.

To compute the numerical fluxes, an algorithm must be provided to compute the wave speeds. The original AUFS proposal used average values from the initial left and right states to approximate s_1 and s_2 . The current technique is to use the results of the acoustic Riemann solver. Thus,

$$s_1 = (u^* - \hat{u}_n)$$

is used for the wave speed and

$$\mathbf{F}_1^c = \mathbf{F}^{lag} = \begin{bmatrix} -u^* \\ 0 \\ n_1 p^* \\ n_2 p^* \\ n_3 p^* \\ p^* u^* \end{bmatrix}$$

is the computed Lagrangian flux. The calculation of the other wave speed depends on whether material is flowing to the left or to the right. For the case when ($s_1 \geq 0$), the flux is determined as

$$s_2 = \min \left(0, \min \left((u_{nL} - \hat{u}_n), (u_{nR} - \hat{u}_n) \right) - \max (c_L, c_R) \right)$$

$$\mathbf{F}_2^c = \begin{bmatrix} 1(u_{nL} - \hat{u}_n - s_2) - u_{nL} \\ \rho_L(u_{nL} - \hat{u}_n - s_2) \\ \rho_L u_{1L}(u_{nL} - \hat{u}_n - s_2) + n_1 p_L \\ \rho_L u_{2L}(u_{nL} - \hat{u}_n - s_2) + n_2 p_L \\ \rho_L u_{3L}(u_{nL} - \hat{u}_n - s_2) + n_3 p_L \\ \rho_L E_L(u_{nL} - \hat{u}_n - s_2) + u_{nL} p_L \end{bmatrix}$$

and if ($s_1 < 0$), then

$$s_2 = \max \left(0, \max \left((u_{nL} - \hat{u}_n), (u_{nR} - \hat{u}_n) \right) + \max (c_L, c_R) \right)$$

$$\mathbf{F}_2^c = \begin{bmatrix} 1(u_{nR} - \hat{u}_n - s_2) - u_{nR} \\ \rho_R(u_{nR} - \hat{u}_n - s_2) \\ \rho_R u_{1R}(u_{nR} - \hat{u}_n - s_2) + n_1 p_R \\ \rho_R u_{2R}(u_{nR} - \hat{u}_n - s_2) + n_2 p_R \\ \rho_R u_{3R}(u_{nR} - \hat{u}_n - s_2) + n_3 p_R \\ \rho_R E_R(u_{nR} - \hat{u}_n - s_2) + u_{nR} p_R \end{bmatrix}$$

These values are then used to compute the normal flux using (2.79).

2.1.12 Boundary Conditions

Because infinitely large systems cannot be considered, the spatial domain must be truncated to include only the portion of interest. Therefore, special treatment is required at the domain boundaries. The current capability provides treatment for outflow and wall boundary conditions.

Boundary conditions for finite-volume methods on unstructured meshes have been discussed extensively in the literature for compressible single-phase flows,

e.g. [53, 60–62]. Berry et al. [63] has a detailed discussion of boundary conditions within the context of DEM. The above references discuss the importance of proper boundary conditions for hyperbolic systems. In order for a simulation to be well-posed, it is critical that the boundaries properly account for the propagation of waves. Outward propagating waves are completely determined by the flow inside the domain, and no boundary conditions can be specified for them. Inward propagating waves, on the other hand, are completely determined by the conditions outside the domain and must be prescribed by boundary conditions. For the DEM, boundary conditions are applied to each phase individually. Therefore, in order for the method to be well-posed, the boundary conditions must meet the same criteria as the original Euler equations.

2.1.12.1 Outflow Boundary Conditions

The outflow boundary condition is used to model far-field boundaries. These boundaries are placed on the edges of the finite domain, across which fluid is allowed to flow. They are characterized by no flow gradients normal to the boundary. This is enforced by assuming that a *ghost* cell exists on the exterior of the domain with the exact same flow properties as the adjacent cell on the interior. This results in

$$\begin{aligned} u^* &= u_{nL} \\ p^* &= p_L \end{aligned} \tag{2.80}$$

where the L state is assumed to be on the interior. The above expressions for u^* and p^* replace the contact velocity and pressure given by (2.75) and (2.76) in the flux calculation. This condition is often referred to as a supersonic outflow condition since information outside the domain cannot influence the state inside. The current implementation uses this condition for both supersonic and subsonic flow. This constrains the boundary such that there are no incoming waves. An enhancement would be to allow a static pressure specification for the case of subsonic outflow, similar to the formulation of Berry et al. [63].

2.1.12.2 Wall Boundary Conditions

The wall boundary condition is used for situations where flow is not allowed to cross the boundary. This condition is often referred to as a slip-wall boundary since material is allowed to slip past the wall in the tangential direction. It is used for external boundaries representing rigid walls, which can be stationary or moving. It can also be used to represent a symmetry plane.

The boundary is enforced by solving a Riemann problem in order to determine the flux. Since no material is allowed to pass through the boundary, the only flux is from the momentum equation in the form of pressure. The contact velocity is prescribed by the boundary condition. Thus, the pressure of the wall can be computed by doing a one-sided Riemann solve using (2.73) to get

$$\begin{aligned} u^* &= \hat{u}_n \\ p^* &= p_L + Z_L (u_{nL} - u^*) \end{aligned} \tag{2.81}$$

where the L state is assumed to be on the interior. The normal mesh velocity, \hat{u}_n , is computed with (2.69) based on the prescribed nodal velocities.

The above condition is similar to the one used by Liou [64], which was derived by integrating the characteristic equation. Liou found that a regular extrapolation condition ($p^* = p_L$) resulted in short-wave oscillations when shocks reflect off the boundary. The technique used above eliminates those oscillations.

This boundary condition is the same one used at Lagrangian boundaries between the fluid and deforming structure. This coupling will be discussed in more detail in a later section.

2.1.13 Second-Order Extension

The solver has been implemented in the context of a traditional MUSCL-Hancock predictor-corrector scheme to achieve second-order accuracy in both time and space. See Toro [53] for a discussion of these types of methods. A primitive variable formulation is used to extrapolate data from the cell centers to the faces. It also advances the solution in time to the mid-level of the time-step. This is

the predictor. The corrector step performs a Riemann solve using these predicted values for the left and right states.

The predictor follows the methodology of Sun and Takayama [58]. To simplify the description of the method, the one-dimensional form of the equations will be presented. The technique can be readily extended to three dimensions. Recalling the primitive variable vector (2.56), the values at the faces are computed by setting

$$\mathbf{w}_f^{n+\frac{1}{2}} = \mathbf{w}_c^n + \frac{\Delta t}{2} \left(\frac{\partial \mathbf{w}}{\partial t} \right)_c^n + \left(x_f^{n+\frac{1}{2}} - x_c^n \right) \left(\frac{\partial \mathbf{w}}{\partial x} \right)_c^n \quad (2.82)$$

where the subscript f indicates that the variable is evaluated at the cell face, while c designates evaluation at the cell centroid. The superscripts indicate that the values are evaluated at time-level n or $(n + \frac{1}{2})$.

Recall that the mesh moves in the current method. Therefore, it is observed that the spatial location of the face, to which the data is being extrapolated, varies with time. Thus, the face's location is evaluated at time $(n + \frac{1}{2})$, which can be computed as

$$x_f^{n+\frac{1}{2}} = x_f^n + \frac{\Delta t}{2} \hat{u}_f$$

with the mesh velocity \hat{u}_f being computed with (2.69). Therefore, the extrapolated coordinate vector is given by

$$\left(x_f^{n+\frac{1}{2}} - x_c^n \right) = \frac{(\Delta x)_c^n}{2} + \frac{\Delta t}{2} \hat{u}_f \quad (2.83)$$

where Δx is the distance between opposite faces of the cell evaluated at the beginning of the time-step.

Recalling (2.55), the time derivative is given by

$$\begin{aligned} \frac{\partial \mathbf{w}}{\partial t} &= -\mathbf{a} \frac{\partial \mathbf{w}}{\partial x} + \mathbf{s} \\ &= -u \frac{\partial \mathbf{w}}{\partial x} - \mathbf{a}^{off} \frac{\partial \mathbf{w}}{\partial x} + \mathbf{s} \end{aligned} \quad (2.84)$$

where $\mathbf{a}^{off} = (\mathbf{a} - uI)$ consists of the off-diagonal terms of matrix \mathbf{a} . Notice that the time derivative is taken with respect to the fixed coordinate system rather

then the moving coordinate system associated with the mesh. This is the required reference frame since (2.82) was derived with the time derivative evaluated at the fixed spatial position of x_c^n . Thus, no contributions from the mesh motion are required in the above expression.

Inserting expressions (2.83) and (2.84) into (2.82) yields

$$\begin{aligned} \mathbf{w}_f^{n+\frac{1}{2}} &= \mathbf{w}_c^n + \left(\frac{\Delta x}{2} - \frac{\Delta t}{2} (u_c - \hat{u}_f) \right) \left(\frac{\partial \mathbf{w}}{\partial x} \right)_c^n \\ &\quad + \frac{\Delta t}{2} \left(\mathbf{a}^{off} \left(\frac{\partial \mathbf{w}}{\partial x} \right)_c^n + s \right) \end{aligned} \quad (2.85)$$

where the gradient terms have been merged and the interpolation to the faces has been combined with the evaluation of the advection terms. A simplified form of the above predictor is

$$\mathbf{w}_f^{n+\frac{1}{2}} = \mathbf{w}_c^n + \left(\frac{\Delta x}{2} - \frac{\Delta t}{2} (u_c - \hat{u}_f) \right) \left(\frac{\partial \mathbf{w}}{\partial x} \right)_c^n \quad (2.86)$$

where the off-diagonal and source terms have been neglected. This modification has proved to be very robust and computationally efficient. It is still second order with respect to advection and improves the accuracy of the material transport, compared with the first-order method. The inclusion of the off-diagonal and source terms was investigated, but it was found that they did not have a noticeable effect on the method's accuracy for any of the attempted calculations. Also, the discretization of these terms is ambiguous because it requires an approximation of the interface velocity and pressure, and it is not clear how to compute these values consistently with DEM. Thus, the form (2.86) is used throughout this study.

It is well understood that second-order linear schemes will produce spurious oscillations near discontinuities (see Toro [53] for discussion). It is desired to find monotone solutions with no oscillations while still attaining second-order accuracy in regions of smooth flow variations. Therefore, non-linear schemes must be used. This is accomplished through the use of van Leer-type slope limiters in the current methodology. The limiter is applied to the gradient term, $\frac{\partial \mathbf{w}}{\partial x}$, in (2.86). The use of slope limiters is discussed in Toro [53], Berger et al. [65], and Barth [66].

The evaluation of the gradient within a computational cell will use the grid stencil shown in Figure 2.5. Dimensional splitting is employed for computing the gradient, such that the horizontal and vertical gradients are computed sequentially. The gradient for cell c in the horizontal direction is calculated by first computing the change in the variable across the cell. This is given by

$$\Delta_{wce} = \frac{\Omega_e \phi_e + \Omega_c \phi_c}{\Omega_e + \Omega_c} - \frac{\Omega_w \phi_w + \Omega_c \phi_c}{\Omega_w + \Omega_c} \quad (2.87)$$

where ϕ represents a component of the primitive variable vector \mathbf{w} and Ω is the volume of the cell. The c subscript indicates that the quantity is associated with the center cell c , while the subscripts w and e indicate that the quantity is associated with the neighbor cell in the *west* and *east* directions respectively. The above expression uses a volume-weighted average to approximate the change in ϕ in the west-east direction. For uniform meshes, the above approximation will exactly reproduce linear gradients, but for stretched meshes the approximation is not exact. This error will always underestimate the slope. Thus, the approximation will add additional diffusion, but the solution will remain monotonic.

Next the changes across the faces are computed

$$s_{wc} = \frac{\phi_c - \phi_w}{\Delta_{wce}} \quad (2.88)$$

$$s_{ce} = \frac{\phi_e - \phi_c}{\Delta_{wce}} \quad (2.89)$$

where s_{wc} and s_{ce} are the relative jumps across the w - c face and the c - e face respectively. These values are then used to compute a scale factor

$$\mathcal{L} = \begin{cases} 0 & \text{if } (s_{wc} \cdot s_{ce} < 0) \\ \max(0, \min(s_{wc}, s_{ce}, 1)) & \text{otherwise.} \end{cases} \quad (2.90)$$

which acts as the slope limiter. The final slope is given by

$$\overline{\frac{\partial \phi}{\partial x}} = \frac{\mathcal{L} \Delta_{wce}}{\Delta x} \quad (2.91)$$

where the bar over the gradient is a reminder that a slope limiter has been applied.

This limited slope is then used in (2.86) to determine the values of ϕ at the *east* and *west* faces. The same technique is subsequently applied in the vertical direction to get the values on the *north* and *south* faces, as well as in the out-of-plane direction for three-dimensional problems. The directional splitting approach requires the use of three-dimensional hex elements or two-dimensional quad elements. Techniques for general element formulations are discussed in Berger et al. [65].

2.1.14 Sub-grid Phase Coupling (Relaxation Processes)

This section discusses the sub-grid coupling between phases. These interactions are often referred to as relaxation processes. These terms are included to account for interactions between phases which are not well resolved within the spatial and temporal discretization. Therefore, these phenomena are accounted for with engineering models.

The discussion begins with descriptions of the models. First is a discussion of pressure relaxation (§2.1.14.1), which accounts for mechanical compression of the phases. This is followed by temperature relaxation (§2.1.14.2), or heat transfer. Finally, the drag model, or velocity relaxation (§2.1.14.3), is explained. At the end is a summary of the relaxation models, including some details about the solution strategy.

2.1.14.1 Pressure Relaxation

Consider a compressible solid particle embedded within a carrier gas. Pressure differences between the two phases will initiate relaxation waves which propagate from the particle surface to the center of the particle, then reflect back to the surface. These waves cause the solid and gas to compress or expand in a struggle to balance the mechanical force acting at the particle surface. This process will edge the phases towards equilibrium. This process is exhibited in the conservation

equations (2.5). Isolating the relevant terms results in

$$\begin{aligned}
\frac{\partial(\alpha_g)}{\partial t} &= -\mu(p_p - p_g) \\
\frac{\partial(\alpha_g \rho_g)}{\partial t} &= 0 \\
\frac{\partial(\alpha_g \rho_g u_g)}{\partial t} &= 0 \\
\frac{\partial(\alpha_g \rho_g E_g)}{\partial t} &= \mu p^I (p_p - p_g) \\
\frac{\partial(\alpha_p \rho_p)}{\partial t} &= 0 \\
\frac{\partial(\alpha_p \rho_p u_p)}{\partial t} &= 0 \\
\frac{\partial(\alpha_p \rho_p E_p)}{\partial t} &= -\mu p^I (p_p - p_g)
\end{aligned}$$

Combining the first and fourth equations results in

$$\frac{\partial(\alpha_g \rho_g E_g)}{\partial t} = -p^I \frac{\partial \alpha_g}{\partial t}$$

As seen in the second and third equations, the mass and momentum are constant.

Therefore, obtaining

$$\frac{\partial \varepsilon_g}{\partial t} = -\frac{p^I}{\alpha_g \rho_g} \frac{\partial \alpha_g}{\partial t}$$

which relates the change in internal energy to the volume change due to the pressure work done on the gas. A similar expression can be obtained for the quantities associated with the particles.

Chinnayya et al. [22] estimates a relaxation time scale for the pressures to equilibrate as

$$\tau_P \sim \frac{d_p}{c_p}$$

where d_p is the particle diameter and c_p is the sound speed of the particle. The pressure-relaxation time scale is generally much smaller than the time scales associated with velocity and temperature. In addition, for particle sizes much smaller than the computational cells used for the discretization of the transport equations, the pressure-relaxation time scale is less than the Courant-limited time-steps given by (2.63). Thus, it is generally acceptable to assume that the phases are in pressure equilibrium.

Therefore, the pressure-relaxation model adjusts the volume fractions of the phases until mechanical equilibrium is reached. A number of pressure-equilibrium models are discussed in the literature (e.g. [57, 67]). The current technique is based on the method of Tipton [68]. This is an iterative technique and is robust for general equations of state.

Discussion of Particle Compaction

Under certain conditions pressure equilibrium is not appropriate. When multiphase mixtures are subjected to mechanical loading, the mixture is compressed. For dilute particle flows, the majority of the compression occurs within the gas phase, i.e. the volume occupied by the gas decreases while the volume of the particles is relatively unchanged. As the loading increases, the gas volume decreases, resulting in less space between particles. Eventually the particles will contact each other, preventing full compression of the material. At this point, resistance to the applied loading is no longer dominated by gas pressure, but rather inter-particle stresses caused by particle-particle interactions. This inter-particle stress is referred to as the compaction stress. This added stress results in non-equilibrium between the gas and particle pressures. Many researchers have discussed the importance of compaction on multiphase dynamics [18, 69–71].

At this time compaction is not included in the current study. However, its future implementation is straightforward. Basically a compaction model (as discussed in the above references) provides a configurational pressure, β_p , that accounts for particle-particle interaction. This added pressure is then used as an offset within the pressure-relaxation algorithm. Thus, the end state is not pressure equilibrium, but rather a state where $p_p - \beta_p - p_g$ is zero.

2.1.14.2 Temperature Relaxation

The temperature-relaxation module accounts for heat transfer between phases. Consider an isolated discrete particle embedded within a continuum fluid. When a temperature difference exists between the particle and the fluid, heat transfer will occur at the material interface going from the hot to the cold material. The heat

flux at the particle surface (q_w) is defined as the heat transfer from the particle to the surrounding fluid divided by the surface area of the discrete particle (A_p). This can be related to the change in temperature of the particle through conservation of energy

$$-A_p q_w = \frac{d\mathcal{E}_p}{dt} = m_p C_{vp} \frac{dT_p}{dt}$$

where \mathcal{E}_p is the extensive thermal energy of the discrete phase. It has been assumed that the mass (m_p) of the particle does not vary during the temperature relaxation step. Therefore, any mass transfer and the associated energy exchange due to chemical reactions or evaporation (both of which are neglected in the current study) would need to be accounted for by the chemistry model. The specific heat (C_{vp}) can be a function of temperature, and it is outside the derivative by definition, i.e. there is no need to assume it is constant at this point.

The heat flux can be characterized by the Nusselt number, a non-dimensional parameter defined as

$$Nu = \frac{q_w d_p}{k_c (T_p - T_c)} \quad (2.92)$$

where T_p is the temperature of the discrete particle, T_c is the temperature of the continuum fluid, k_c is the thermal conductivity of the gas, and d_p is the particle diameter. Combining the definition of the Nusselt number with the conservation of energy (see Crowe et al. [4]) results in an evolution equation for the temperature of an isolated particle

$$\begin{aligned} \frac{dT_p}{dt} &= \frac{Nu}{2} \frac{12k_c}{\rho_p C_{vp} d_p^2} (T_c - T_p) \\ &= \frac{1}{\tau_T} (T_c - T_p) \end{aligned} \quad (2.93)$$

where the density of the particle, ρ_p , has been combined with its volume to simplify the equation and the thermal response time is given by

$$\tau_T = \frac{\rho_p C_{vp} d_p^2}{12k_c} \frac{2}{Nu}$$

The above equations neglect radiative heat transfer. They also assume that the temperature is uniform throughout the interior of the particle, which is a valid assumption as long as the Biot number for the flow is less than 0.1.

Continuum based multiphase models, which utilize a continuum representation of both the continuous and discrete phases, are not interested in the behavior of a single isolated particle. Rather, within a single computational cell, each discrete phase is assumed to be a cloud of particles. Fortunately, even though (2.93) has been derived for an isolated particle, it is easily seen that the equation is also valid for a particle cloud where each particle shares the same properties and temperature. This assumes that each particle is thermally isolated from other particles and thus all heat transfer occurs between isolated particles and the surrounding gas. Therefore, (2.93) can be applied to each particle phase, such that

$$\frac{dT_\pi}{dt} = \frac{1}{\tau_\pi} (T_0 - T_\pi) \quad (2.94)$$

where the subscript $\pi = 1, N$ designates one of the N particle phases. Although the multiphase model allows for multiple gas phases, the thermal-relaxation model treats them all as a combined bulk-gas phase, denoted with the subscript 0. The determination of the bulk-gas properties is discussed in §D.2. Note that there is a thermal response time associated with each particle phase and the subscript T has been dropped from τ for simplicity.

Closure of the above system of equations, (2.94), is obtained by using conservation of energy over all particle and gas phases. Combining (2.94) with the definition of the extensive thermal energy given above results in expressions for the change in total energy of each particle phase

$$\frac{d\mathcal{E}_\pi}{dt} = \frac{m_\pi C_{v\pi}}{\tau_\pi} (T_0 - T_\pi)$$

Assuming that there is no heat transfer or work being done across the boundary of the computational cell, conservation of energy yields

$$\frac{d\mathcal{E}_0}{dt} + \sum_{\pi=1}^N \left[\frac{d\mathcal{E}_\pi}{dt} \right] = 0$$

resulting in

$$m_0 C_{v0} \frac{dT_0}{dt} + \sum_{\pi=1}^N \left[\frac{m_\pi C_{v\pi}}{\tau_\pi} (T_0 - T_\pi) \right] = 0$$

$$\frac{dT_0}{dt} = - \sum_{\pi=1}^N \left[\frac{c_\pi}{\tau_\pi} (T_0 - T_\pi) \right] \quad (2.95)$$

where

$$c_\pi = \frac{m_\pi C_{v\pi}}{m_0 C_{v0}} = \frac{\alpha_\pi \rho_\pi C_{v\pi}}{\alpha_0 \rho_0 C_{v0}}$$

Thus, the system is closed and the final set of equations is

$$\frac{dT_0}{dt} = - \left(\sum_{\pi=1}^N \frac{c_\pi}{\tau_\pi} \right) T_0 + \sum_{\pi=1}^N \left[\frac{c_\pi}{\tau_\pi} T_\pi \right] \quad (2.96)$$

$$\frac{dT_\pi}{dt} = \frac{1}{\tau_\pi} T_0 - \frac{1}{\tau_\pi} T_\pi \quad : \pi = 1, N$$

The initial conditions for the above system are given by the phase temperatures at the beginning of the thermal-relaxation stage, T_π^{old} . The system is solved to get the updated temperature, T_π^{new} , using the methods discussed in §D. The time between the initial and updated state is the time-step Δt . After the new temperatures have been computed, the phase internal energy is updated

$$\Delta \varepsilon_\pi = C_{v\pi} (T_\pi^{new} - T_\pi^{old}) \quad (2.97)$$

to account for the temperature change.

2.1.14.3 Velocity Relaxation

The velocity-relaxation module accounts for momentum transfer between phases, primarily drag. Consider an isolated particle embedded within a continuum fluid. Due to force interactions between the particle and the fluid, momentum transfer will occur. Momentum conservation is governed by Newton's second law

$$m_p \frac{d\vec{u}_p}{dt} = \vec{F} = \vec{F}_D + \vec{F}_P + \dots \quad (2.98)$$

where the total force, \vec{F} , is the result of aerodynamic drag and the pressure variations within the fluid. Additional forces, such as the virtual mass or Basset force, could be included. Assuming they are constant over a single time-step would allow them to be included in the source term associated with the pressure gradient. However, at this time, the additional terms have not been investigated. Gravity and other body-force contributions to the force are not included here as they are more appropriately accounted for within the hydrodynamic phase. Crowe et al. [4] discusses the terms contributing to particle force in detail.

The drag force is assumed to obey the relation

$$\vec{F}_D = \frac{1}{2} \rho_c V_R^2 \frac{(\vec{u}_c - \vec{u}_p)}{V_R} C_D \frac{\pi d_p^2}{4} \quad (2.99)$$

where \vec{u}_p and \vec{u}_c are the particle and continuum fluid velocity respectively, ρ_c is the fluid density, d_p is the particle diameter, $V_R = |\vec{u}_c - \vec{u}_p|$ is the relative velocity of the particle, and C_D is the drag coefficient. The above expression for the aerodynamic drag results directly from the definition of the drag coefficient.

Pressure gradients within the continuum fluid give rise to a force in the direction of the gradient. Assuming the pressure gradient is constant in the region around the particle surface, the net force acting on the particle is given by

$$\vec{F}_P = -\frac{m_p}{\rho_p} \nabla p_c$$

where ρ_p is the particle density and p_c is the fluid pressure.

Plugging the above force definitions into (2.98) yields the governing equation for particle motion

$$\begin{aligned} \frac{d\vec{u}_p}{dt} &= \frac{3}{4} \frac{\rho_c}{\rho_p} \frac{C_D}{d_p} V_R (\vec{u}_c - \vec{u}_p) - \frac{\nabla p_c}{\rho_p} \\ &= \frac{1}{\tau_V} (\vec{u}_c - \vec{u}_p) - \vec{q}_p \end{aligned} \quad (2.100)$$

where the velocity response time is given by

$$\tau_V = \frac{4}{3} \frac{\rho_p}{\rho_c} \frac{d_p}{C_D V_R}$$

and the non-homogeneous source term is

$$\vec{q}_p = -\frac{\nabla p_c}{\rho_p}$$

Continuum based multiphase models, which utilize a continuum representation of both the continuous and discrete phases, are not interested in the behavior of a single isolated particle. Rather, within a single computational cell, each discrete phase is assumed to be a cloud of particles. Fortunately, even though (2.100) has been derived for an isolated particle, it is easily seen that the equation is also valid for a particle cloud where each particle shares the same properties and velocity. This assumes that each particle is isolated from other particles and thus all force interactions occur between isolated particles and the surrounding gas. Therefore, (2.100) can be applied to each particle phase, such that

$$\frac{d\vec{u}_\pi}{dt} = \frac{1}{\tau_\pi} (\vec{u}_0 - \vec{u}_\pi) - \vec{q}_\pi \quad (2.101)$$

where the subscript $\pi = 1, N$ designates one of the N particle phases. Although the multiphase model allows for multiple gas phases, the velocity-relaxation model treats them all as a combined bulk-gas phase, denoted with the subscript 0. The determination of the bulk-gas properties is discussed in §D.2. Note that there is a velocity response time associated with each particle phase and the subscript V has been dropped from τ for simplicity.

Closure of the above system of equations, (2.101), is obtained by using conservation of momentum over all particle and gas phases. Combining (2.101) with the definition of momentum results in expressions for the change in total momentum of each particle phase

$$\frac{d(m_\pi \vec{u}_\pi)}{dt} = \frac{m_\pi}{\tau_\pi} (\vec{u}_0 - \vec{u}_\pi) - m_\pi \vec{q}_\pi$$

Assuming that there is no force across the boundary of the computational cell, conservation of momentum yields

$$\frac{d(m_0 \vec{u}_0)}{dt} + \sum_{\pi=1}^N \left[\frac{d(m_\pi \vec{u}_\pi)}{dt} \right] = 0 \quad (2.102)$$

resulting in

$$m_0 \frac{d\vec{u}_0}{dt} + \sum_{\pi=1}^N \left[\frac{m_\pi}{\tau_\pi} (\vec{u}_0 - \vec{u}_\pi) \right] + \sum_{\pi=1}^N [m_\pi \vec{q}_\pi] = 0$$

$$\frac{d\vec{u}_0}{dt} = - \sum_{\pi=1}^N \left[\frac{c_\pi}{\tau_\pi} (\vec{u}_0 - \vec{u}_\pi) \right] - \sum_{\pi=1}^N [c_\pi \vec{q}_\pi] \quad (2.103)$$

where

$$c_\pi = \frac{m_\pi}{m_0} = \frac{\alpha_\pi \rho_\pi}{\alpha_0 \rho_0}$$

Thus, the system is closed and the final set of equations is

$$\frac{d\vec{u}_0}{dt} = - \left(\sum_{\pi=1}^N \frac{c_\pi}{\tau_\pi} \right) \vec{u}_0 + \sum_{\pi=1}^N \left[\frac{c_\pi}{\tau_\pi} \vec{u}_\pi \right] - \sum_{\pi=1}^N [c_\pi \vec{q}_\pi] \quad (2.104)$$

$$\frac{d\vec{u}_\pi}{dt} = \frac{1}{\tau_\pi} \vec{u}_0 - \frac{1}{\tau_\pi} \vec{u}_\pi + \vec{q}_\pi$$

The initial conditions for the above system are given by the phase velocities at the beginning of the velocity-relaxation stage, \vec{u}_π^{old} . The system is solved to get the updated velocity, \vec{u}_π^{new} , using the methods discussed in §D. The time between the initial and updated state is the time-step Δt .

Finally, after the new velocities have been computed, the phase internal energy is updated. This comes from conservation of total energy. Since the kinetic energy of the phases has changed, the internal energy must be adjusted appropriately to conserve total energy. Therefore, the internal energy is updated such that

$$\frac{d\varepsilon_0}{dt} = \frac{d}{dt} \left(\frac{\vec{u}_0 \cdot \vec{u}_0}{2} \right) + \sum_{\pi=1}^N \left[c_\pi \frac{d}{dt} \left(\frac{\vec{u}_\pi \cdot \vec{u}_\pi}{2} \right) \right] \quad (2.105)$$

$$\frac{d\varepsilon_\pi}{dt} = 0 \quad (2.106)$$

where the choice was made to dump all the excess energy into the gas phase. Therefore, viscous heating will tend to heat the gas. Dividing up the energy to all phases using a mass weighting has been tried, but it is believed that heating the gas is more realistic for the majority of applications.

2.1.14.4 Summary of Relaxation Solution

The previous sections (§2.1.14.2 and §2.1.14.3) derived models for energy and momentum transfer between phases. Both relaxation models result in similar systems of ODE's which govern the interactions. Therefore, the governing equations for both models can be solved using the same numerical techniques. Two distinct solution strategies have been developed to solve these equations for use in the current method. The first is an analytic solution method where a closed-form solution is generated using an eigenvalue decomposition. The analytic method is exact for the case of constant coefficients. For non-constant coefficients the method drops to first order. The second method solves the equations numerically using the backward-Euler time-stepping scheme. This method is also first-order accurate. These solution algorithms are discussed in detail in §D.

As discussed in §2.1.9, the relaxation processes are handled using an operator-split approach. Therefore, within a time-step, the hydrodynamic conservation laws are solved first. The relaxation terms are solved afterwards. Each relaxation process is solved separately in a serial sequence: first velocity relaxation, followed by temperature relaxation, and then pressure relaxation. Details of the steps are discussed in the following table.

Velocity Relaxation

The velocity relaxation operator is the first to be evaluated. It is applied by looping over every cell in the multiphase fluid and performing the following calculations:

- Evaluate the equation of state and material properties of the discrete phases as well as the bulk gas (see §D.2) within the cell.
- The local Reynolds number of the particles is determined using the relative motion between the particle and gas phases. This is used to compute a drag coefficient for each discrete phase. A number of drag coefficient formulations are available, which are outlined in §B. The

drag coefficient is used to evaluate the velocity response time, τ_V .

- All of the coefficients in (2.104) have now been calculated, so these equations are solved using either the analytic or backward-Euler integration method. This is done for each coordinate direction. This provides updated velocity vectors for each phase.
- The internal energy is updated to account for the change in kinetic energy by integrating (2.105).

Temperature Relaxation

The temperature relaxation operator is evaluated next. It is applied by looping over every cell in the multiphase fluid and performing the following calculations:

- Evaluate the equation of state and material properties of the discrete phases as well as the bulk gas (see §D.2) within the cell.
- A Nusselt number is computed for each discrete phase. A variety of Nusselt number formulations are available, which are outlined in §C. The Nusselt number is used to evaluate the thermal response time, τ_T .
- All of the coefficients in (2.96) have now been calculated, so these equations are solved using either the analytic or backward-Euler integration method. This provides updated temperature values for each phase.
- The internal energy is updated to account for the change in thermal energy using (2.97).

Pressure Relaxation

The pressure relaxation operator is the last to be evaluated. Pressure relaxation is fundamentally different than temperature and velocity relaxation. An iterative scheme is applied to each computational cell until pressure equilibrium is achieved:

- Apply a change to the volume fraction.
- Update the density and internal energy due to the volume change.
- Evaluate the pressure of each phase using its equation of state.
- If pressure equilibrium has been reached, quit. If not, do another iteration.

It is important to note that the initial conditions for each step are a result of the previous stage. Therefore, the initial conditions for the drag model come from the hydrodynamic solve. Initial conditions for the temperature relaxation come from the velocity relaxation. The pressure equilibrium starts out with the results of the temperature relaxation. Also note that it is possible for subsequent operations to undo previous results. For example, for very small thermal response times the temperature relaxation model will predict thermal equilibrium between the phases. However, the pressure-relaxation module subsequently updates the temperature and the phases will not be in temperature equilibrium anymore. Thus, the model is not equivalent to a temperature-pressure equilibrium model.

2.2 The Structural Model

This section discusses the structural mechanics model. It begins with a presentation of the governing equations. This is followed by an outline of the algorithm used to solve them.

The current research has focused primarily on the development of the multiphase capability. The structural model was taken from an existing hydrocode and required very little modification. Therefore, the documentation of the structural model is much less detailed than the previous description of the multiphase model. The reader is referred to [40] for additional theoretical documentation of the structural model.

2.2.1 Lagrangian Equations

The governing equations for the structural material are given by the Euler equations (2.1–2.3). However, unlike the earlier discussion which presented the conservation laws in an Eulerian frame, the structural model uses a Lagrangian representation. These equations can be found in numerous mechanics texts (e.g. [72]). Defining ρ^0 as the initial material density and \mathcal{J} as the determinant of the deformation gradient, the Lagrangian equations are given by

Mass Balance

$$\rho \mathcal{J} = \rho^0 \quad (2.107)$$

Momentum Balance

$$\rho \frac{D(u_i)}{Dt} = \frac{\partial(\sigma_{ij})}{\partial x_j} \quad (2.108)$$

Energy Balance

$$\rho \frac{D(\varepsilon)}{Dt} = \sigma_{ij} D_{ij} \quad (2.109)$$

The rate of deformation is given by

$$D_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$$

and the Cauchy stress tensor, σ_{ij} , includes the deviatoric component representing material strength. The material derivatives, $\frac{D}{Dt}$, move with the material.

2.2.2 Lagrangian Algorithm

The governing equations (2.107–2.109) are solved using an explicit Lagrangian methodology. The scheme is second-order in both space and time. The method is staggered in space. This means that the velocity field is centered at the nodes and uses tri-linear basis functions, while the thermodynamic state (i.e. density, energy, pressure, etc.) is cell-centered and uses piecewise-constant basis functions. It is also staggered in time, where the velocity is evaluated at the mid-level of the

time-step. It is similar to the staggered-grid hydro scheme outlined in Benson [23]. The stress gradients and strain rates for the Lagrangian step are evaluated by a low-order finite-element method (FEM), as in Goudreau and Hallquist [73]. A lumped (diagonal) mass matrix is used. The energy is handled similar to the method of Wilkins [74], where total-energy conservation is not explicitly enforced. This averts the problem of converting lost kinetic energy into internal energy and overheating materials. The pressure work is integrated using a third-order Runge-Kutta method. Shocks are treated with a scalar artificial viscosity and hour-glass modes are damped using an hour-glass correction.

The following procedure is used to update the structural state from time t^n to time t^{n+1} :

- Construct the force at each node at t^n . This involves integrating the surface forces over a control volume at each mesh node. The surface forces have contributions from the material stress, pressure, artificial viscosity, and hour-glass forces.
- Compute the nodal acceleration by dividing the nodal force by the mass of the control volume around the node.
- Integrate the acceleration to get the nodal velocity at $t^{n+\frac{1}{2}}$.
- Integrate the velocity to determine the nodal displacement. This is then used to update the nodal positions at t^{n+1} .
- Evaluate the material state:
 - Evaluate the constitutive model for the strength of the material.
 - Update the artificial viscosity and hour-glass forces.
 - Advance the internal energy based on the work done during the time-step.
 - Evaluate the pressure from the equation of state.

- Advance the time and return to the beginning of the step.

2.3 The Fluid-Structure Interaction Model

The interaction between the multiphase material and the structure is handled at the material interface. A diagram of the computational cells near a material interface is seen in Figure 2.6. As seen in the figure, the structural and multiphase zones share nodes and faces along the interface. This is required for the current method as the interface is required to coincide with the cell boundaries. Thus, the interface must remain Lagrangian.

The overall time-stepping scheme uses an operator-split methodology. The time-step begins by solving the Lagrangian motion of the structure. This moves the structural nodes as well as the nodes on the interface. This is followed by a mesh-relaxation step which moves the multiphase nodes, allowing for a quality mesh for the fluid solution while continuing to conform to the moving structure. The multiphase hydrodynamics is then computed on the moving mesh. Multiphase relaxation and source terms are added at the end. The following sections discuss each of these algorithmic steps individually.

2.3.1 Stage 1: Fluid to Structure Coupling

The first stage in the time-step algorithm is to update the position of the structural nodes using the FEM staggered-grid hydro algorithm discussed in §2.2. During the Lagrangian structural calculation, the multiphase interface is treated as a Neumann (natural) boundary condition. Thus, the fluid imparts force on the structure via surface tractions, but the nodal motion of the interface is determined by the structural FEM method.

Figure 2.6 designates the multiphase zones adjacent to the material interface as *ghost zones*. The ghost zones contribute to the force on the structure. Although the ghost zones are not physically part of the structure, they are treated as structural zones during the FEM calculation and contribute pressure force as well as mass to the structural nodes. The properties in the ghost zones are determined using the

mixture properties of the multiphase material, given by

$$p_{mix} = \sum_{\pi} \alpha_{\pi} p_{\pi} \quad (2.110)$$

$$\rho_{mix} = \sum_{\pi} \alpha_{\pi} \rho_{\pi} \quad (2.111)$$

where the volume-fraction weighted sum is taken over each phase (π) and p_{mix} and ρ_{mix} are the mixture pressure and density, respectively.

A couple of points about the above model should be highlighted. First, the force acting on the structure is determined entirely by the pressures in the gas and particle phases. Thus the particle phase behaves like a fluid and any momentum transferred from the particles to the structure is due to an increase in pressure due to the stagnation of the particle field. This is a fundamentally different process from the manner in which a discrete phase would actually interact with a structure. Discrete particles rely on particle-wall collisions to transmit momentum. This happens through a series of small impulses acting on the structure due to the particles impacting and bouncing off the wall. Despite this disparity, it is noted that the interaction conserves momentum for the system and any deficiencies are expected to be limited to higher-order effects. The second point is that the mass of the gas and particle phases in the ghost cells is included in the structural response. Therefore, the inertia of the multiphase mixture is included in the structural response. Therefore, this is slightly different from applying a pressure boundary condition on the surface of the structure.

2.3.2 Stage 2: Fluid Mesh Relaxation

After the structural finite-element calculation, the next stage is to rezone the multiphase fluid mesh. This stage repositions the internal multiphase nodes which are not on the fluid-structure interface. These nodes are identified by the yellow circles in Figure 2.6. The purpose of this step is to allow the fluid mesh to conform to the moving structural boundary while preserving a smooth mesh for the multiphase calculation and avoiding mesh tangling.

This stage is further separated into two steps. The first step (step 1) moves the nodes in the direction of the material motion. This is used to determine the initial conditions for the second part, which is a mesh smoothing step (step 2).

In step 1, the nodal positions are moved in the direction of the flow. The flow direction is determined from the velocity of the bulk multiphase mixture, given by (D.9). Since the flow velocities are cell-centered, they must be relocated to the nodes. This is done using a modified version of the cell-centered Lagrangian approach of Maire [75]. The implementation of this scheme in the current method only updates the nodal positions and does not modify the material state. Therefore, the method is not consistent with the actual Lagrangian motion of the material. However, since this is just a preprocessing step used to obtain the initial nodal positions for the mesh relaxer, the accuracy of the step is inconsequential.

Step 2 smooths out the mesh. Any mesh-smoothing procedure that retains the original mesh structure and prevents the mesh from tangling could be used with the current ALE scheme. An equipotential relaxation method [76, 77] is generally used. This is an iterative method based on elliptic grid-generation techniques. A mesh generator would typically use hundreds of iterations to converge to a smooth mesh. In the current code, each time-step typically performs a single iteration, although additional iterations can also be specified.

The iterative technique uses the nodal positions computed in step 1 as initial conditions. This is very effective for many applications. For example, forcing the mesh to track the material allows for better resolution of flow features, such as shocks. It has also proved useful for preventing tangling near structural interfaces. However, in some cases the mesh behaves better using the original position. Therefore, there is an option to undo step 1, and revert back to the original position to initialize the mesh relaxer.

The above mesh-relaxation procedure is automated and useful for general applications. However, in some cases it is desired to have a little more control over the exact motion of the mesh. Therefore, the ability to prescribe the mesh

motion has also been included. This option requires the user to specify the nodal positions explicitly at the start of the calculation. This option has been used in many of the test cases presented later in this document.

Once the new mesh locations are determined, the volume fluxes (Ω_f^{n+1}) of the faces are computed. These are calculated geometrically by determining the volume swept out by the face as it moves from its original position at the beginning of the time-step (t^n) to its final position at the end of the relaxation step (t^{n+1}). This volume is shown in Figure 2.2. The volume flux is then used to compute the mesh velocity of the face using (2.69).

2.3.3 Stage 3: Structure to Fluid Coupling

In the final stage, the DEM method is used to compute the multiphase material on the moving mesh using the methods of §2.1. The multiphase model sees the material interface as a wall boundary condition, where no flux passes through the wall. Since this is applied in the normal direction and there is no shear stress in the fluid, this is equivalent to a slip boundary condition for the velocity (see §2.1.12.2). A single-sided Riemann solve (2.81) is performed using the wall velocity and the state in the zone on the multiphase side of the interface. The wall velocity is computed using (2.69), in much the same way that the mesh velocity is computed for all of the other fluid faces. As the mesh velocity is equal to the material velocity, the resulting flux returned from the Riemann solver is the Lagrangian flux. This flux is then used to evaluate the convective flux associated with the DEM method (2.37).

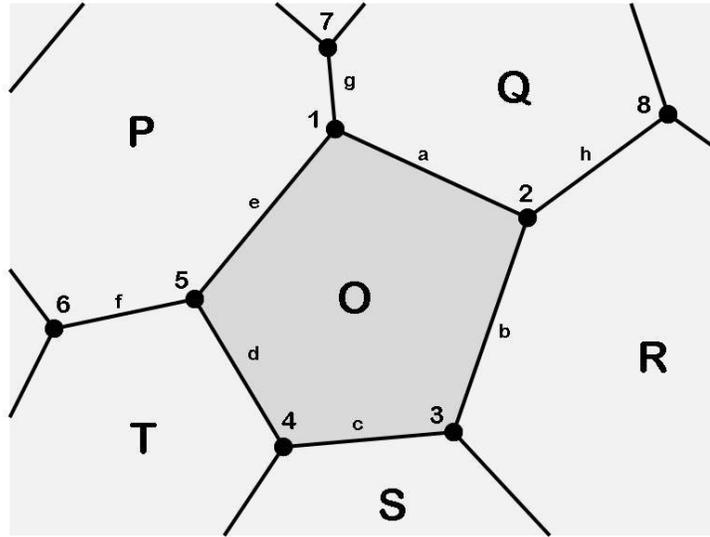


Figure 2.1: Diagram of computational cells. Zones are notated by capital letters, nodes with numbers, faces by lower case letters.

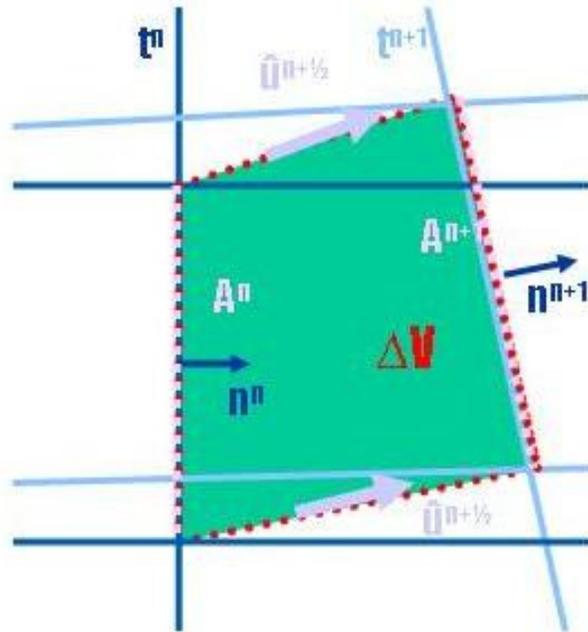


Figure 2.2: Diagram of the volume flux through a moving face. The shaded area represents the volume which is swept out as the face moves. This volume is used to satisfy the geometric conservation laws.

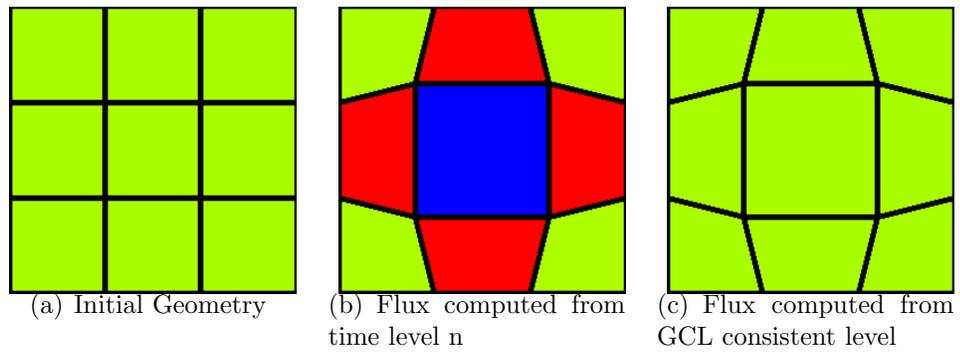


Figure 2.3: An example problem revealing the importance of satisfying the GCL. The colors represent density magnitude.

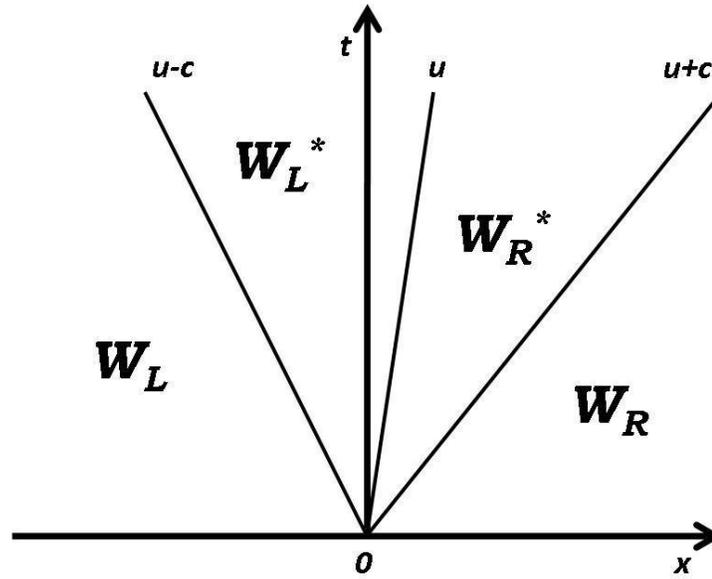


Figure 2.4: Characteristic fan for the Riemann problem

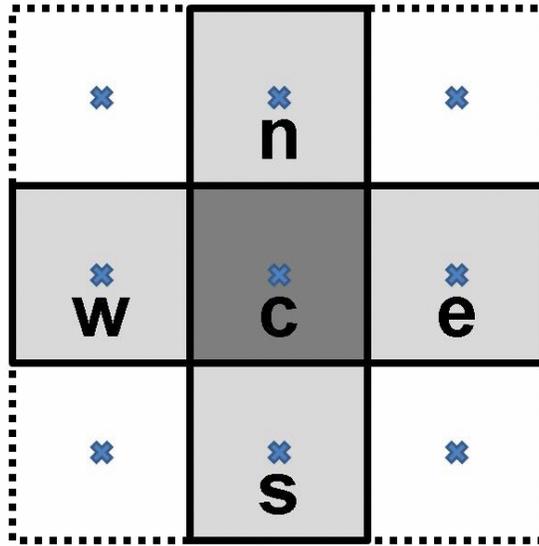


Figure 2.5: Grid stencil used for second-order predictor.

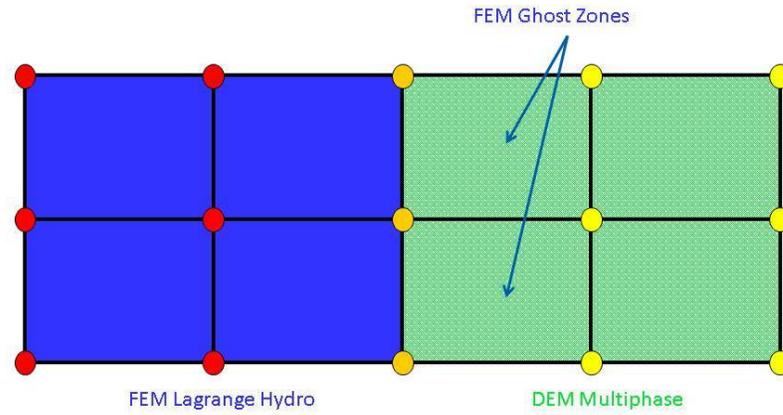


Figure 2.6: Mesh diagram of the multiphase/Lagrangian-structure interface. The green zones are controlled by the multiphase algorithm; the blue zones are controlled by the staggered-grid FEM.

Contact	Surface
1-1	$S_{11} = S_f \min(\alpha_{1,in}, \alpha_{1,out})$
1-2	$S_{12} = S_f \max(0, \alpha_{1,in} - \alpha_{1,out})$
2-1	$S_{21} = S_f \max(0, \alpha_{2,in} - \alpha_{2,out})$
2-2	$S_{22} = S_f \min(\alpha_{2,in}, \alpha_{2,out})$

Table 2.1: Definitions of surface contacts used for computing the convective and Lagrangian fluxes.

Contact	Phase
1-1	$\Psi_{1,11}^* = 1$
1-2	$\Psi_{1,12}^* = \begin{cases} 1 & \text{if } \hat{u}_{12}^* > 0 \\ 0 & \text{otherwise} \end{cases}$
2-1	$\Psi_{1,21}^* = \begin{cases} 1 & \text{if } \hat{u}_{21}^* < 0 \\ 0 & \text{otherwise} \end{cases}$
2-2	$\Psi_{1,22}^* = 0$

Table 2.2: Definitions of inter-cell phase function used for computing the convective fluxes.

Contact	Jump
1-1	$[[\Psi_1]]_{11}^* = 0$
1-2	$[[\Psi_1]]_{12}^* = \begin{cases} -1 & \text{if } \hat{u}_{12}^* < 0 \\ 0 & \text{otherwise} \end{cases}$
2-1	$[[\Psi_1]]_{21}^* = \begin{cases} +1 & \text{if } \hat{u}_{21}^* < 0 \\ 0 & \text{otherwise} \end{cases}$
2-2	$[[\Psi_1]]_{22}^* = 0$

Table 2.3: Definitions of inter-cell jump functions used for computing the Lagrangian fluxes.

Chapter 3

Results

The multiphase model discussed in this document has been exercised on a number of test problems. First and foremost, the test problems are designed to verifying that the model has been implemented correctly. This is done primarily by running tests with known analytic solutions. Some of the tests have been developed to show that the model is applicable to multiphase applications of interest. These are generally compared to experimental test data.

The numerical results have been divided into three sections. The first section tests fundamentals of the numerical implementation on canonical single-phase test problems. The next section looks at simulations involving multiple phases. Finally, the last section exercises the fluid-structure interaction model by simulating multiphase fluids interacting with solid structures.

3.1 Single Phase

The first set of tests will investigate the behavior of the multiphase methodology when it is applied to a single-phase material. In the limit of a single-phase, the current multiphase algorithm reduces down to a system of conservation laws which are equivalent to the Euler equations. The CFD community has developed a large number of canonical test problems for the Euler equations. These tests are commonly used to evaluate the numerical properties of computational models. The tests incorporate flow features which are relevant to both single-phase and multiphase flows, such as shock and rarefaction waves. Therefore, a lot can be learned by comparing single-phase simulations to benchmark solutions of the Euler equations.

The tests begin with a convergence study on the numerical method. This is followed by simulations of the Riemann problem in the Toro and Sod shock-tube problems. Next, an explosive blast is modeled in the Sedov test and an implosion is modeled in the Noh test.

These tests are used to assess the order of accuracy of the model and its stability properties. The behavior of the model near discontinuities is also observed. The application of boundary conditions can also be observed using the single-phase tests. Since the tests were restricted to those which have analytical solutions, the problems are all one-dimensional. However, the Sedov and Noh problems are computed using two-dimensional and three-dimensional meshes, thus utilizing the multi-dimensional capabilities of the model.

3.1.1 Single-Phase Smooth Test Problem

Test Description

The following test is used to verify the order of accuracy of the method for a single-phase material. It is a numerical exercise and not designed to model any physical reality. One of the implicit assumptions when discussing the order of a numerical method is that the solution is *smooth*. Therefore, this test uses smooth initial conditions. Chosen appropriately, smooth spatial variations at the beginning of the run will result in transient behavior that remains smooth throughout the simulation. Simulations are performed on a series of grids in order to compute the convergence rate of the method. An example of a similar test in the literature is found in Schwendeman et al. [48].

Test Setup

This test is designed to exercise the single-phase Euler equations. The phase material uses an ideal-gas equation of state with $\gamma = 1.4$. The material state is initialized using the functions

$$\begin{aligned}\varepsilon &= 2.5 \\ p &= 5.0 + 0.5 \tanh(10x - 4) \\ u &= 0.2 + 0.3 \tanh(20x - 10)\end{aligned}$$

which are smooth on the domain of interest. The functional form of the initial conditions is somewhat arbitrary, but they were chosen in order to obtain smoothly varying transient behavior.

The problem is designed to be one dimensional. It is modeled using a three-dimensional mesh with a single zone in the y and z directions and symmetry boundary conditions on the sides, resulting in a pseudo-1D flow in the x direction, with $x \in [0, 1]$. The number of zones in the x direction are varied to look at mesh convergence. Outflow boundary conditions are given at the boundaries $x = 0$ and $x = 1$. An Eulerian mesh, with no mesh motion, is used for the simulations.

The solutions are compared at the simulation end time $t = 0.05$. A constant

time-step was used for these tests. The size of the time-step was chosen to be $\Delta t = \frac{0.05}{N}$, where N is the number of grid cells. This was found to satisfy the stability conditions at all times.

Test Theory

No analytic solution is available for this test. Smooth analytic solutions are difficult to obtain in general. However, the exact solution can be approximated with sufficient accuracy using the current numerical method with a very large number of grid cells. For the current study 12,800 grid cells were used to obtain the baseline solution. Baseline solutions were obtained using both the first-order and the second-order algorithms. The first-order solution is used as the baseline for the first-order results on the coarser meshes, and the second-order solution is used as the baseline for the second-order results.

Test Results

Solutions were obtained using both the first-order and second-order schemes. The first-order (predictor off) results are plotted in Figure 3.1, and the second-order (predictor on) results are plotted in Figure 3.2. Calculations using four mesh-refinement levels were performed using 100, 200, 400, and 800 grid cells. The numerical results for the various grid resolutions are given by the colored symbols, while the *high-resolution* solution is designated as *Theory* and represented by the solid black line.

All of the calculations resulted in a smooth final flow-field with no shocks or contact discontinuities developing. The results also reveal spatial variation in all of the plotted variables. It appears that the calculations are converging towards an answer as the mesh is refined. The first-order solutions exhibit more variation between the finest and coarsest grids than the second-order method.

Test Error Quantification and Convergence

The discretization error for the solution can be quantified using the fine-grid solution as a baseline. The error (E_N) is determined for a mesh-refinement level N using the methods of §E. The calculated error for the current solutions has been

computed for the variables density, velocity, pressure, and internal energy.

The errors are tabulated along with the associated convergence rates (r_N) in Table 3.1 for the first-order method. The results indicate that the first-order scheme is indeed first-order since the convergence rates are very close to one.

The errors for the second-order method are shown in Table 3.2. The convergence rate for a second-order scheme should have a value of two. However, the current results point toward a convergence rate closer to one, thus indicating first-order convergence rather than second-order. It is noted that the magnitude of the error is lower for the second-order method than for the first-order.

Two possible explanations for not attaining second-order have been identified. First, it is noted that the error reported in this test is a combination of both spatial and temporal discretization error. Therefore, it is possible that the current algorithm is second-order in space, but only first-order in time. Second, despite the smooth flow profile, it is possible that the slope limiters used in the predictor step are influencing the convergence behavior of the second-order scheme.

In order to investigate this further, an additional series of tests have been run with a couple changes. In the previous test the time-step was reduced as the mesh refinement was increased. In the new tests, the time-step was given a constant value of $\Delta t = \frac{0.05}{12800}$ for all grid levels. Therefore, the time-discretization error should be nearly constant throughout all grid levels. Also, the new tests were run with the predictor on, but the slope limiters were disabled. The observed error and convergence rates for these new tests are tabulated in Table 3.3. As seen, second-order behavior has been restored.

It should be noted that just disabling the slope limiters, while keeping the time-steps the same as in the earlier tests does not exhibit second-order convergence rates. Therefore, the dropping in order is probably due to errors in the time discretization. This indicates that the observed order of accuracy for the method is second order in space, but only first order in time.

It is not currently understood why the time-stepping scheme fails to

demonstrate second-order convergence. A possibility is that the use of the simplified predictor (2.86) is causing the rate of convergence to drop. Perhaps using the full predictor (2.85) would regain the desired order of accuracy. Resolving this issue will be an important component of future research.

Test Conclusions

A test problem utilizing smooth initial conditions has been used to assess the order of accuracy of the current method for a single-phase flow. The following conclusions can be drawn from this calculation:

- The current first-order scheme demonstrates an observed convergence rate of first-order.
- The default second-order scheme does not demonstrate second-order convergence, but rather exhibits convergence rates closer to first-order. Disabling the slope limiters in the scheme appear to regain second-order convergence in space. However, the time-stepping scheme still appears to be only first order in time.
- The current test relies on one-dimensional geometries. It would be straightforward, although more computationally expensive, to extend the current test to three dimensions.
- A more rigorous (and more complicated) technique for order-of-accuracy verification is the method of manufactured solutions as discussed by Roache [78] and Salari et al. [79]. The idea here is to manufacture an exact solution and then modify the original governing equations by adding source terms in such a way that the manufactured function becomes the exact solution. This could be a more systematic way to quantify the scheme's accuracy.

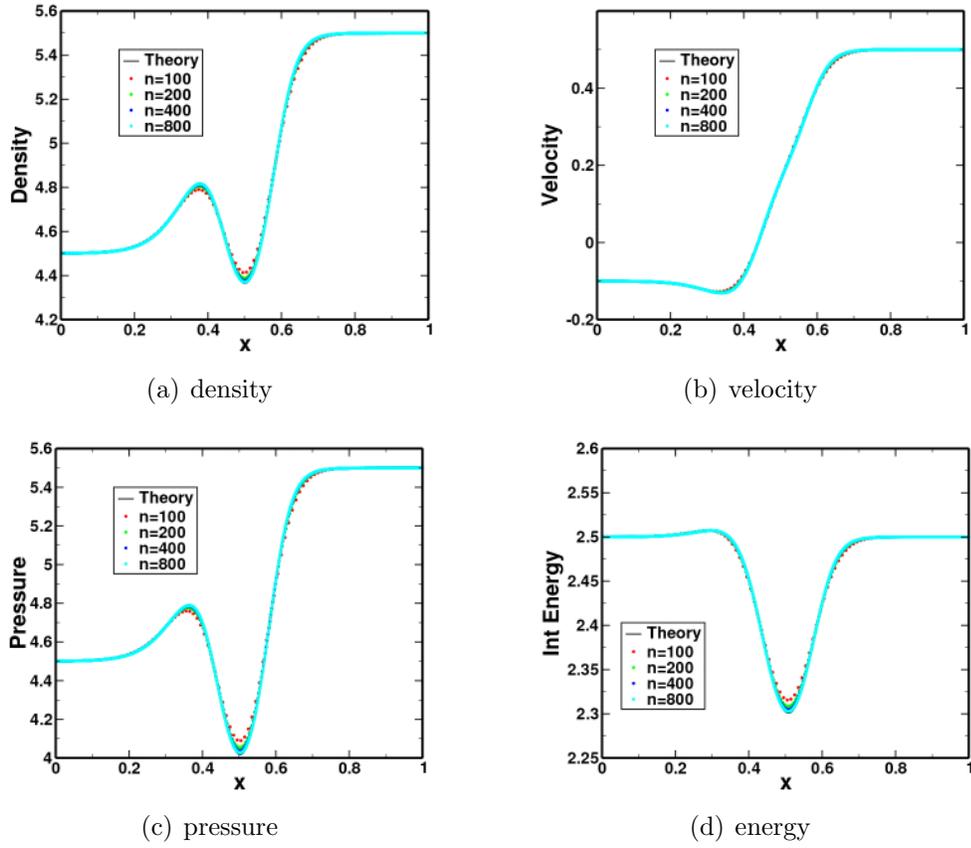


Figure 3.1: Numerical results for the single-phase smooth test. The tests used an *Eulerian grid* with the *predictor off*.

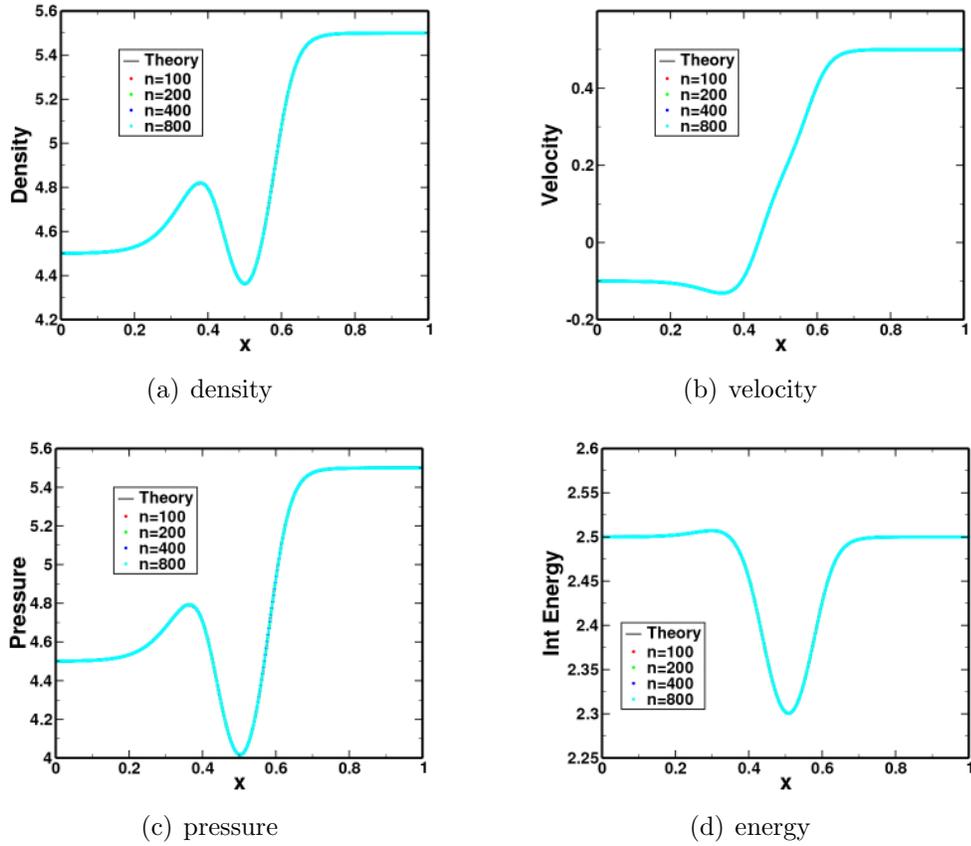


Figure 3.2: Numerical results for the single-phase smooth test. The tests used an *Eulerian grid* with the *predictor on*.

N	E_N	r_N
Density		
100	1.95e-02	
200	1.01e-02	0.94
400	5.13e-03	0.98
800	2.52e-03	1
Velocity		
100	3.74e-03	
200	1.99e-03	0.91
400	1.02e-03	0.97
800	5.01e-04	1
Pressure		
100	2.76e-02	
200	1.44e-02	0.94
400	7.26e-03	0.98
800	3.56e-03	1
Energy		
100	4.99e-03	
200	2.59e-03	0.94
400	1.31e-03	0.98
800	6.43e-04	1

Table 3.1: Numerical error and convergence rates for the single-phase smooth test. The tests used an *Eulerian grid* with the *predictor off*.

N	E_N	r_N
Density		
100	1.39e-03	
200	6.26e-04	1.1
400	3.08e-04	1
800	1.53e-04	1
Velocity		
100	3.91e-04	
200	1.42e-04	1.5
400	6.39e-05	1.2
800	3.10e-05	1
Pressure		
100	1.89e-03	
200	8.75e-04	1.1
400	4.35e-04	1
800	2.16e-04	1
Energy		
100	2.94e-04	
200	1.49e-04	0.99
400	7.71e-05	0.95
800	3.94e-05	0.97

Table 3.2: Numerical error and convergence rates for the single-phase smooth test. The tests used an *Eulerian grid* with the *predictor on*.

N	E_N	r_N
Density		
100	1.08e-03	
200	2.55e-04	2.1
400	6.19e-05	2
800	1.54e-05	2
Velocity		
100	2.53e-04	
200	6.36e-05	2
400	1.61e-05	2
800	4.08e-06	2
Pressure		
100	1.48e-03	
200	3.48e-04	2.1
400	8.47e-05	2
800	2.11e-05	2
Energy		
100	2.22e-04	
200	5.27e-05	2.1
400	1.32e-05	2
800	4.16e-06	1.7

Table 3.3: Numerical error and convergence rates for the single-phase smooth test. The tests used an *Eulerian grid* with the *predictor on/no limiter*.

3.1.2 Toro Test Problems

Test Description

The Riemann problem is an initial-value problem for hyperbolic conservation laws where the initial conditions consist of piecewise constant data having a single discontinuity. The Riemann problem is very useful for understanding the Euler equations because all properties, such as shocks, rarefaction waves, and contact discontinuities, appear as characteristics in the solution. Thus, exact solutions can be obtained for comparison to numerical calculations.

The text by Toro [53] specifies a number of numerical tests given by the Riemann problem with various initial conditions. The tests were selected to test the performance of numerical methods for wave patterns likely to be encountered in typical hydrodynamic problems.

Test Setup

Seven test problems were investigated as defined by [53, p 329]. These problems look at the single phase Euler equations for an ideal gas with $\gamma = 1.4$. The tests consist of two constant states, $\{\rho_L, u_L, p_L\}$ on the left and $\{\rho_R, u_R, p_R\}$ on the right. The spatial domain of the problem is $x \in [0, 1]$ where the left and right states are separated by a discontinuity at position $x = x_0$. The solutions are compared at the simulation end time $t = t_0$. The initial conditions for the seven tests are tabulated in Tables 3.4–3.5.

The problem is one dimensional by definition. It is modeled using a three-dimensional mesh with a single zone in the y and z directions and symmetry boundary conditions on the sides, resulting in a pseudo-1D flow in the x direction. The number of zones in the x direction are varied to look at mesh convergence. Outflow boundary conditions are given at the boundaries $x = 0$ and $x = 1$. An Eulerian mesh, with no mesh motion, is used for the simulations.

Test Theory

There is no closed-form solution for the Riemann problem of the Euler equations. However, exact solutions can be obtained using iterative procedures. The exact

solution of the Riemann problem is computed using the methods discussed in Toro [53].

Test Results

The numerical results of the seven tests are plotted in Figures 3.3–3.16. The numerical results for the various grid resolutions are given by the colored symbols, while the exact solution is represented by the solid black line. The computations were performed on a fixed Eulerian mesh using both the first-order method (predictor off) and the second-order scheme (predictor on). The current simulations using the AUFS Riemann solver, exhibit similar behavior as the results for the first-order HLLC approximate Riemann solver as presented in [53].

Test 1 (Figure 3.3) is a modified version of the Sod problem [80] (see §3.1.3). It consists of a rightward traveling shock, a rarefaction wave on the left, and in the middle there is a contact discontinuity traveling to the right following the shock. The first-order numerical approximation exhibits some numerical diffusion, as the shock has been smeared over 7 or 8 grid cells. As expected, the contact is even more diffusive. The rarefaction seems to be reasonably resolved with only minor diffusion occurring at the head and tail of the wave. The calculation appears to be monotonic in the vicinity of the shock and contact as there are no spurious oscillations for this test case. It appears that the average position of the waves is correct, indicating that speed of propagation of the waves is accurately captured in the calculations. Turning the predictor on (Figure 3.10) reduces the diffusion in the shock and contact. Small oscillations can be seen at the location of the contact in the velocity and pressure curves.

Test 2 (Figure 3.4) is referred to as the *123 problem* and consists of two symmetric rarefaction waves and a stationary contact. The region between the rarefaction waves is close to vacuum, which leads to difficulties in numerical schemes. That can be seen in the calculation of the internal energy. The density and pressure are computed reasonably. However, due to their small magnitude, small errors in the density manifest as much larger errors in the energy.

Test 3 (Figure 3.5) is the left half of the blast problem of Woodward and Colella [81]. It is a very strong (Mach 198) shock on the right, a contact and a rarefaction on the left. The calculations show diffusion in the contact, resulting in low values of density after the shock. The first-order solution does appear monotone and it seems to be converging to the exact solution as the grid is refined. The results with the predictor on (Figure 3.12) show improved resolution of the discontinuities. However, non-monotonic oscillations can be seen at the shock and the rarefaction.

Test 4 (Figure 3.6) represents the collision of two shocks. It consists of a right traveling shock, a right traveling contact, and another shock on the left which is also traveling slowly to the right. The contact is very diffusive in this case, although both the left and right shocks are very sharply resolved. The results with the predictor (Figure 3.13) are generally well-behaved, but show small wiggles in the region between the shocks.

Test 5 (Figure 3.7) tests the ability of the method to capture very slowly-moving contact discontinuities. It involves a right shock, a left rarefaction, and a stationary contact. It is similar to test 3, but the contact is stationary. The contact is very sharply resolved in the calculations resulting in very good representation of the post-shock state. However, there are some non-monotonic overshoots at the shock, and even more-pronounced overshoots at the contact. The overshoots increase when the predictor is turned on (Figure 3.14) and also show up at the tail of the rarefaction.

Tests 6 and 7 consist of an isolated contact discontinuity. In Test 6 (Figure 3.8) the contact is stationary. In Test 7 (Figure 3.9) the contact is moving slowly to the right. Notice that the current computations resolve the contact discontinuity exactly for test 6, thus exhibiting similar behavior as the HLLC solver. The resolution of the shock is improved when the predictor is turned on (Figure 3.16), with no overshoots seen in this case.

Test Conclusions

The Toro test problems provide an excellent framework for analyzing the accuracy of a numerical method and the performance of the Riemann solver for common wave structures encountered in typical hydrodynamic problems. The following conclusions can be drawn from these calculations:

- The calculations show that the current code using the AUSF Riemann solver can be used to accurately represent hydrodynamic phenomena.
- The first-order scheme is fairly diffusive near shocks and contact surfaces, as would be expected for first-order methods.
- The second-order predictor scheme is less diffusive near shocks and contact discontinuities. However, unphysical oscillations are present in some of the tests. These oscillations are similar to the ones observed in Toro [53] for MUSCL-Hancock schemes. There it is observed that modifying the slope limiter could potentially reduce the oscillations at the expense of additional diffusion. This should be investigated with the current scheme.

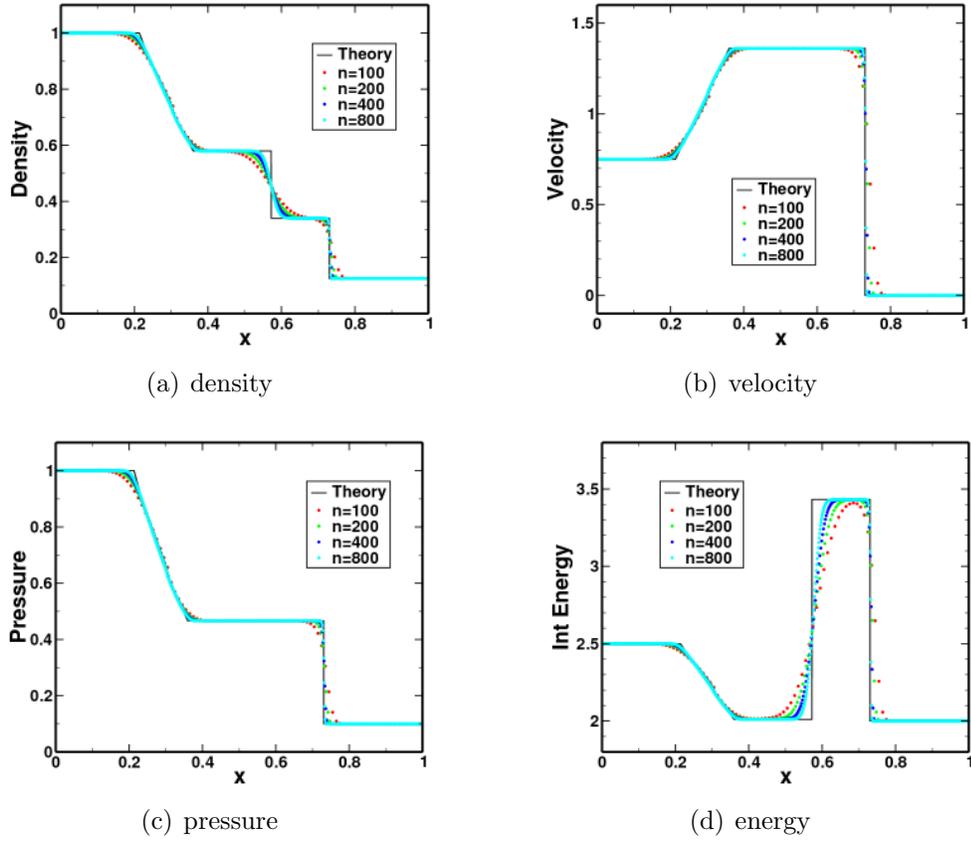


Figure 3.3: Single-phase numerical results for Toro Test #1. The tests used an *Eulerian grid* with the *predictor off*.

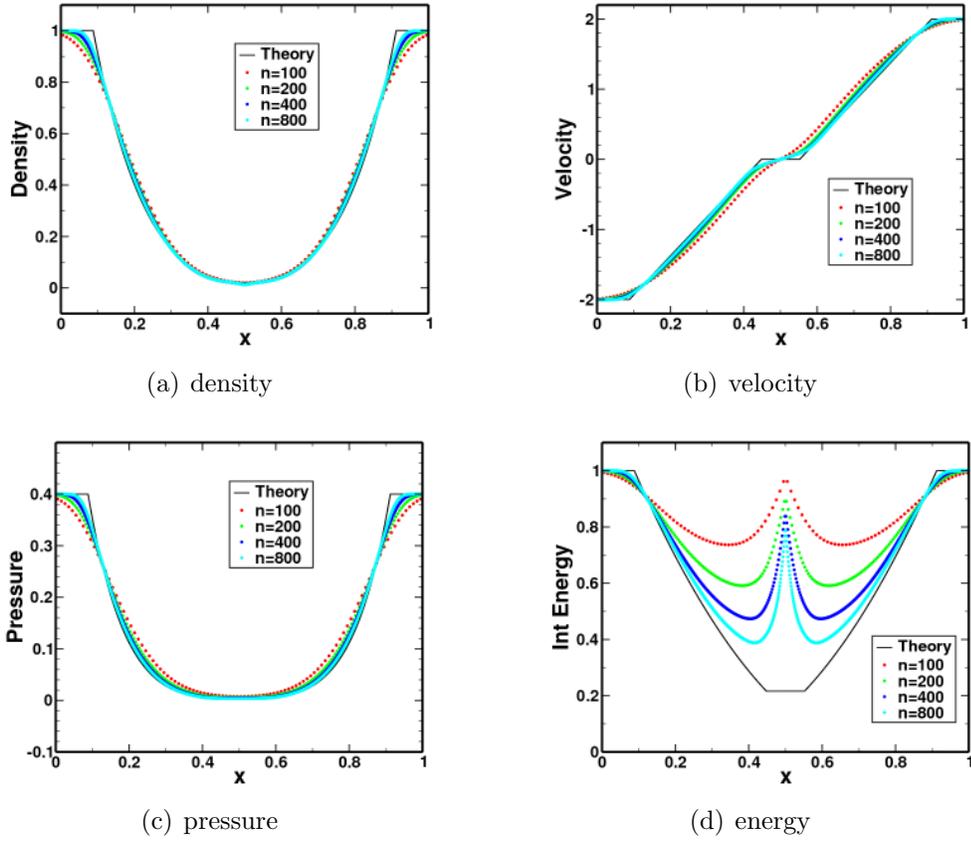


Figure 3.4: Single-phase numerical results for Toro Test #2. The tests used an *Eulerian grid* with the *predictor off*.

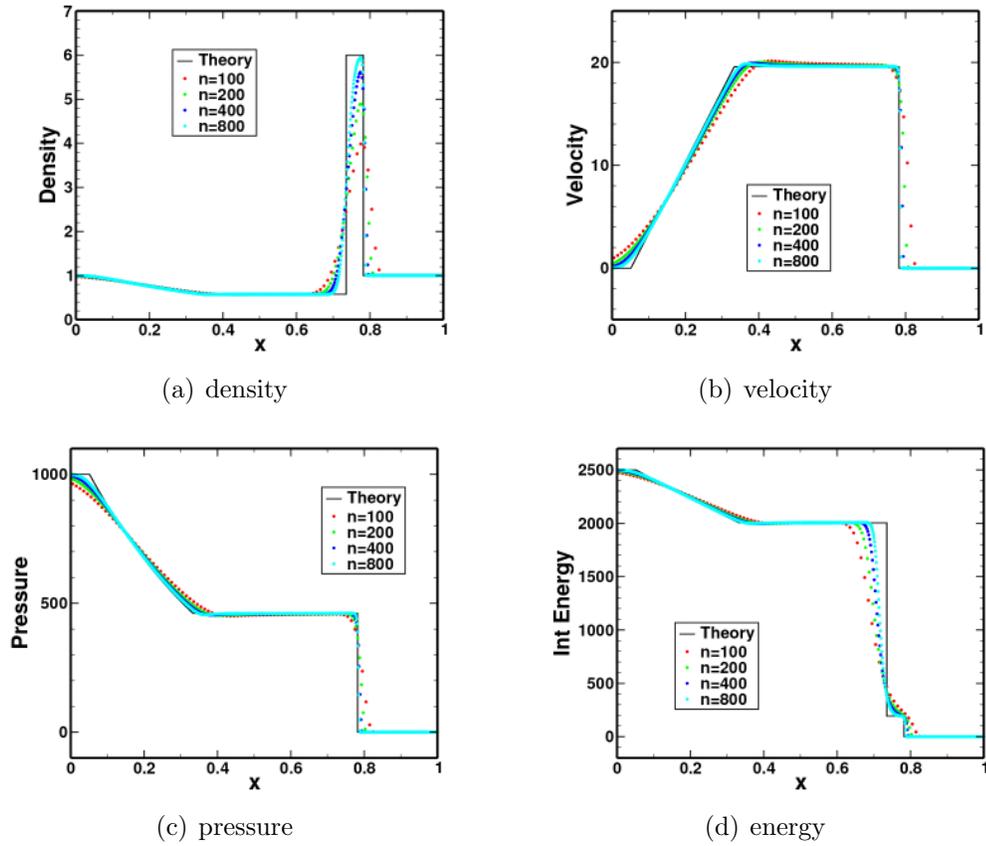
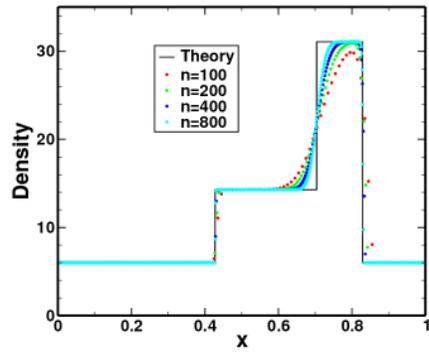
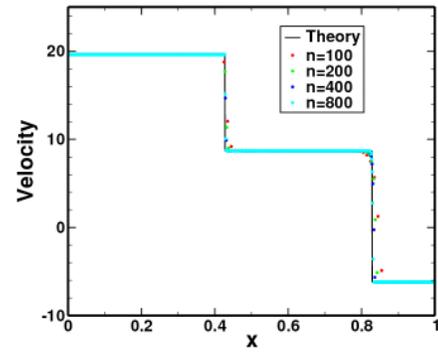


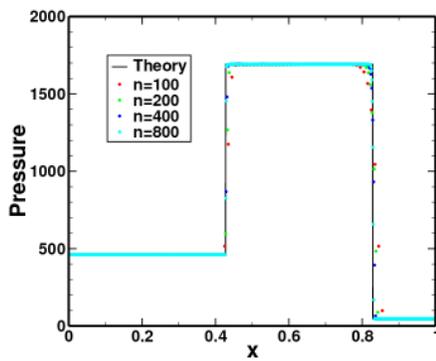
Figure 3.5: Single-phase numerical results for Toro Test #3. The tests used an *Eulerian grid* with the *predictor off*.



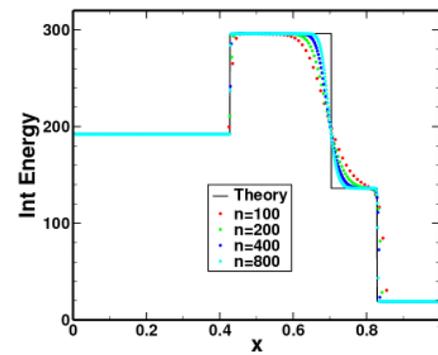
(a) density



(b) velocity



(c) pressure



(d) energy

Figure 3.6: Single-phase numerical results for Toro Test #4. The tests used an *Eulerian grid* with the *predictor off*.

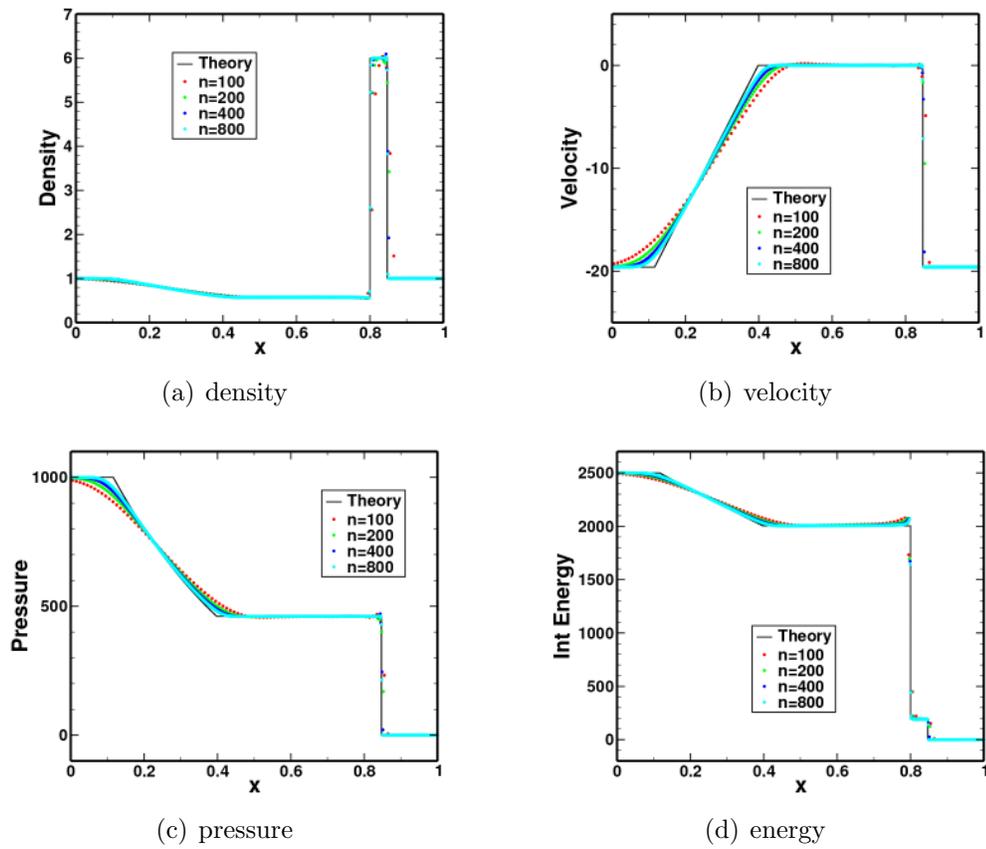


Figure 3.7: Single-phase numerical results for Toro Test #5. The tests used an *Eulerian grid* with the *predictor off*.

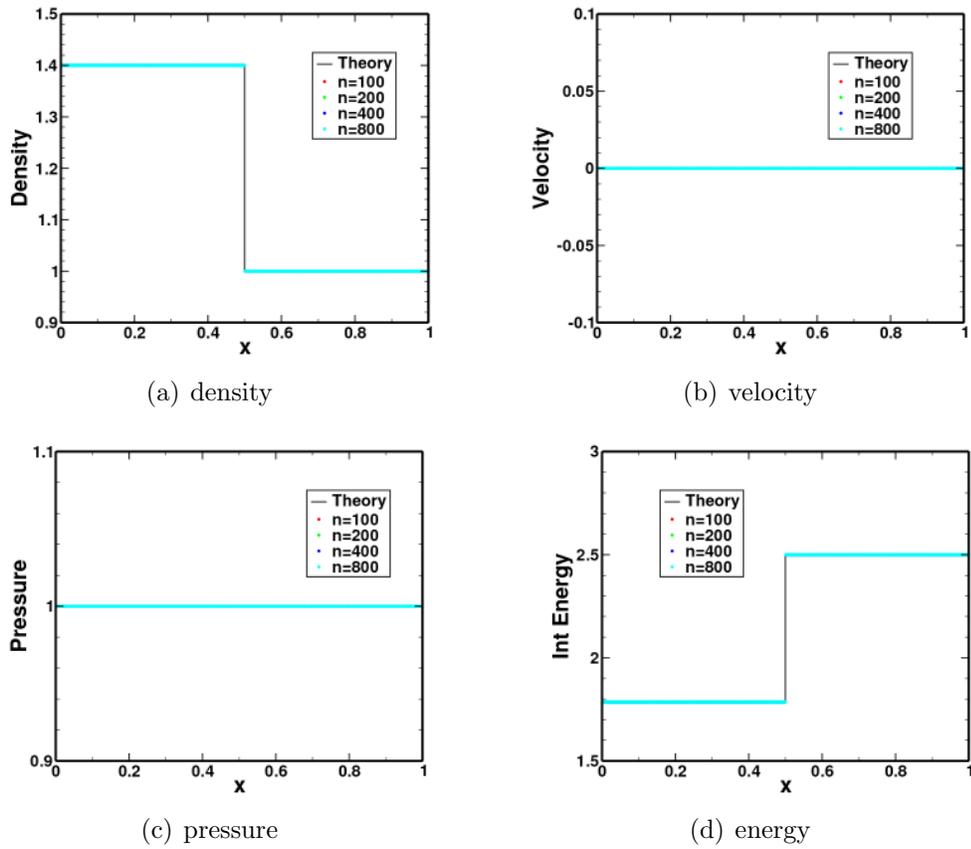


Figure 3.8: Single-phase numerical results for Toro Test #6. The tests used an *Eulerian grid* with the *predictor off*.

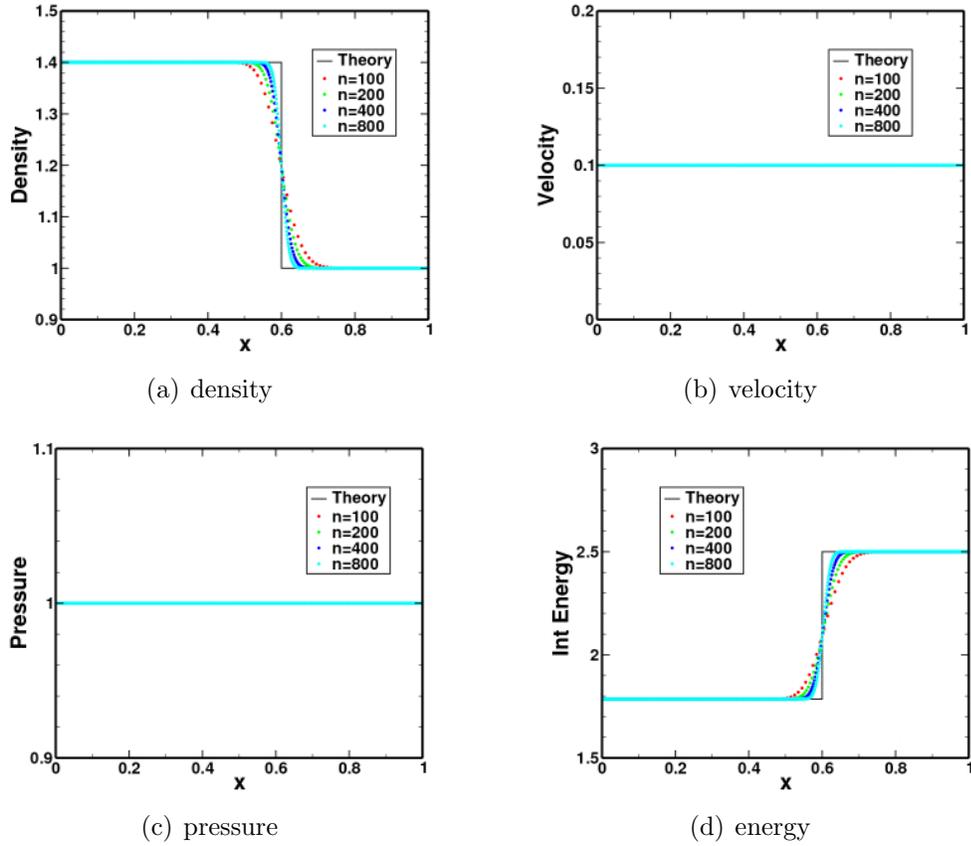


Figure 3.9: Single-phase numerical results for Toro Test #7. The tests used an *Eulerian grid* with the *predictor off*.

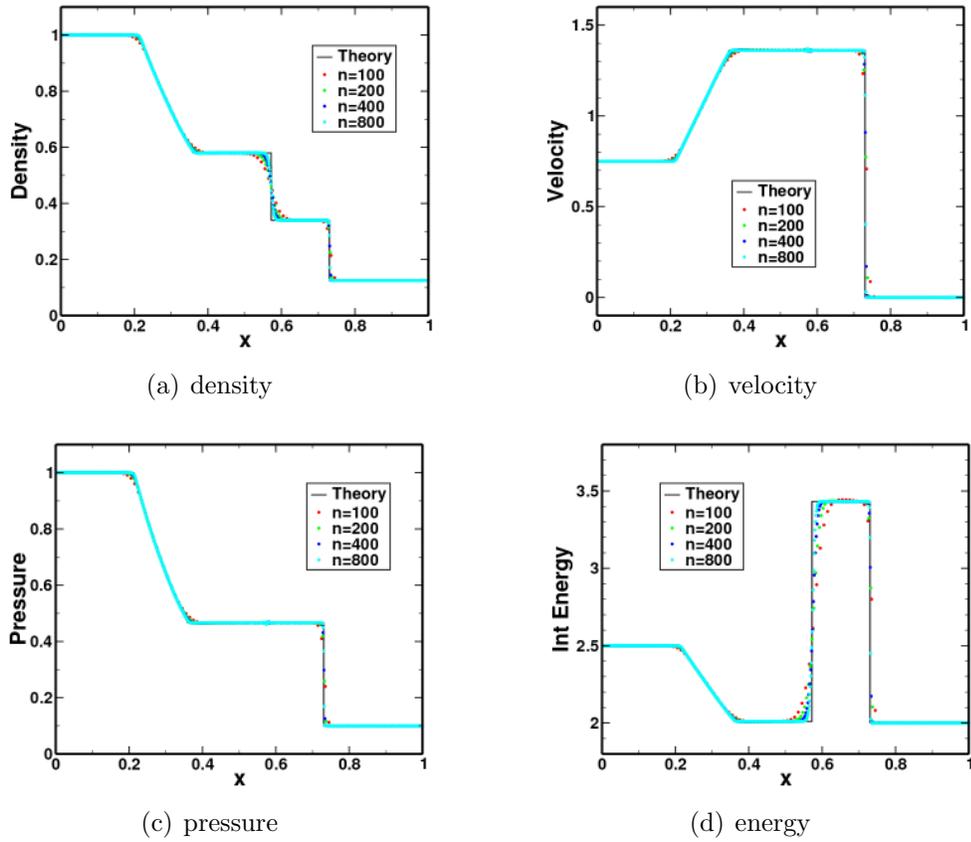


Figure 3.10: Single-phase numerical results for Toro Test #1. The tests used an *Eulerian grid* with the *predictor on*.

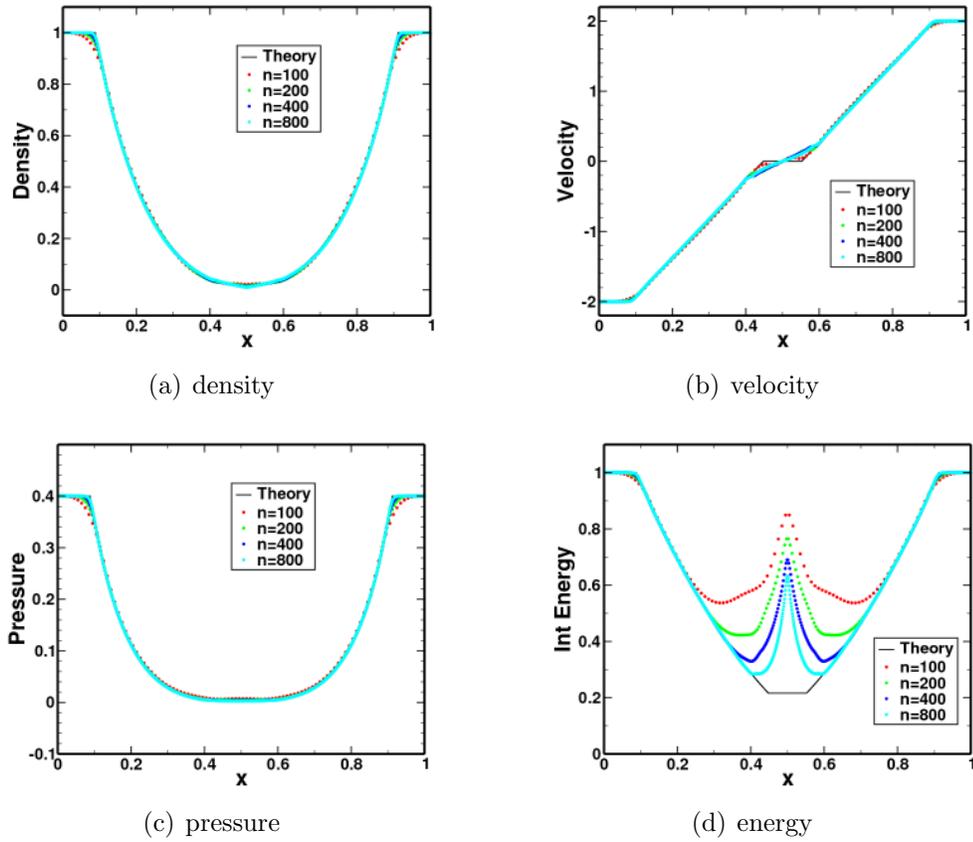


Figure 3.11: Single-phase numerical results for Toro Test #2. The tests used an *Eulerian grid* with the *predictor on*.

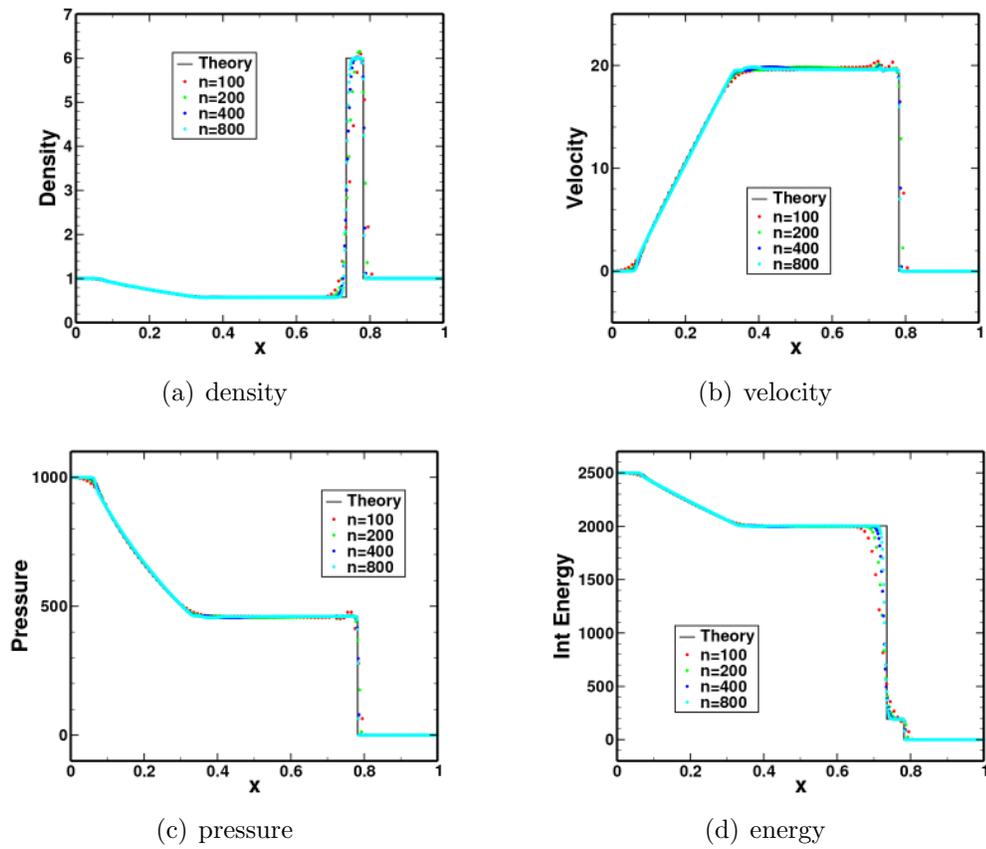


Figure 3.12: Single-phase numerical results for Toro Test #3. The tests used an *Eulerian grid* with the *predictor on*.

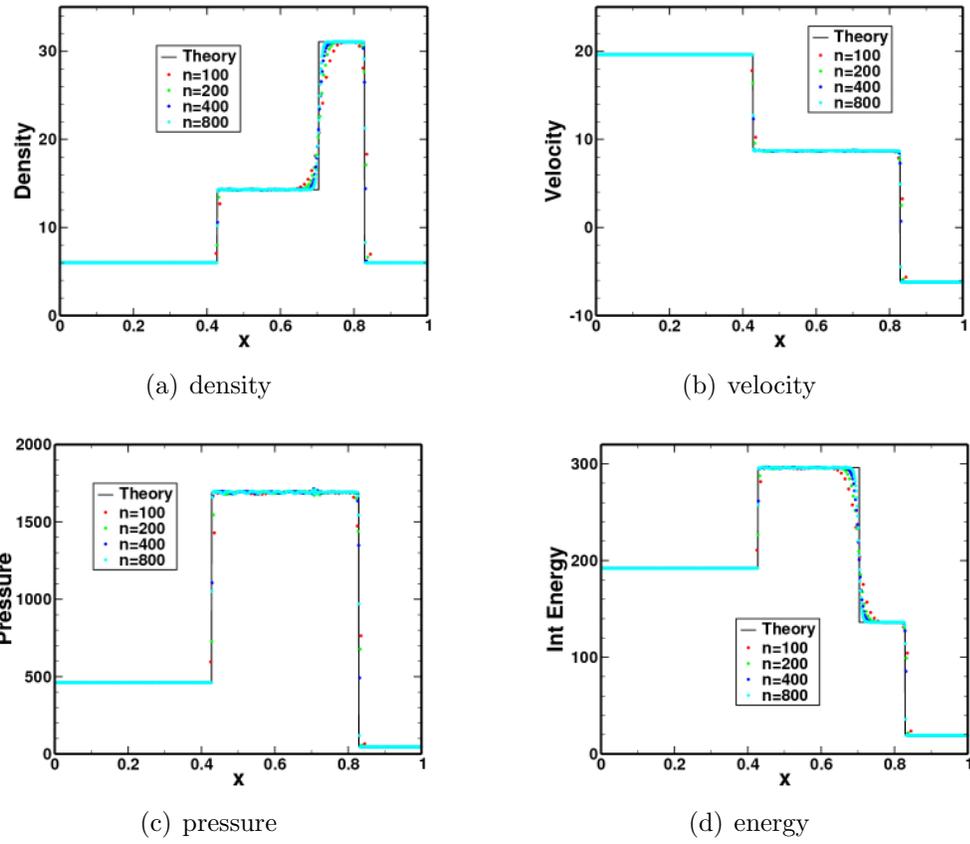


Figure 3.13: Single-phase numerical results for Toro Test #4. The tests used an *Eulerian grid* with the *predictor on*.

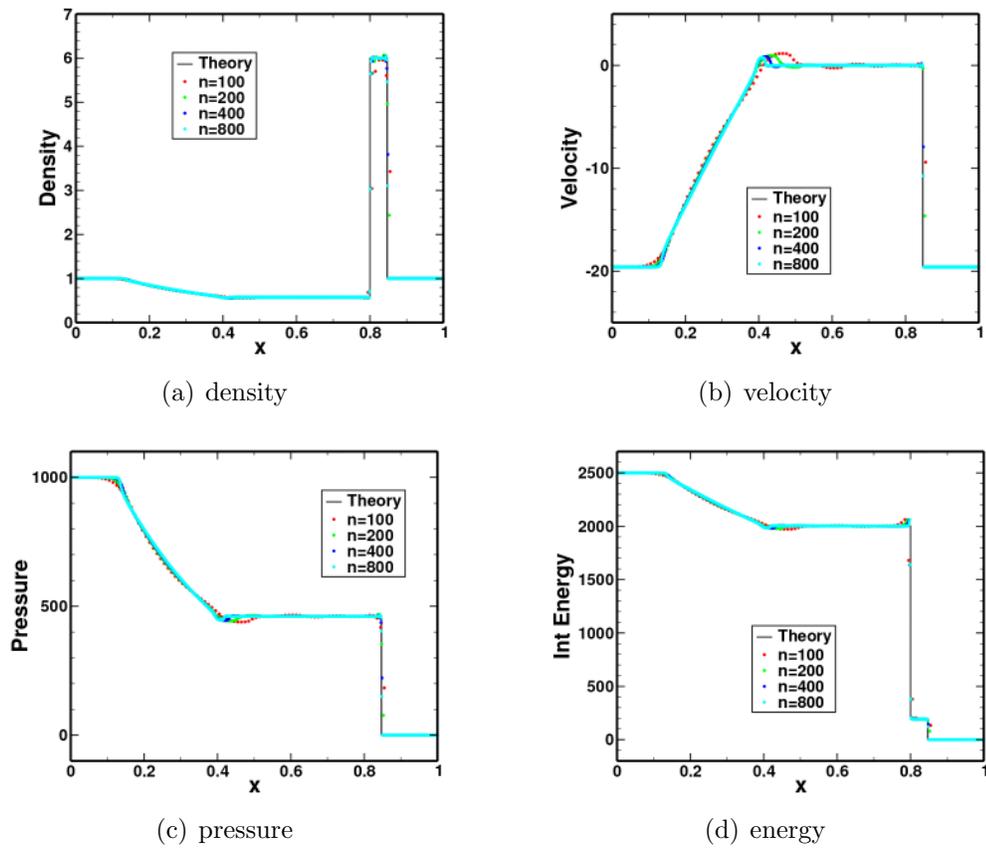


Figure 3.14: Single-phase numerical results for Toro Test #5. The tests used an *Eulerian grid* with the *predictor on*.

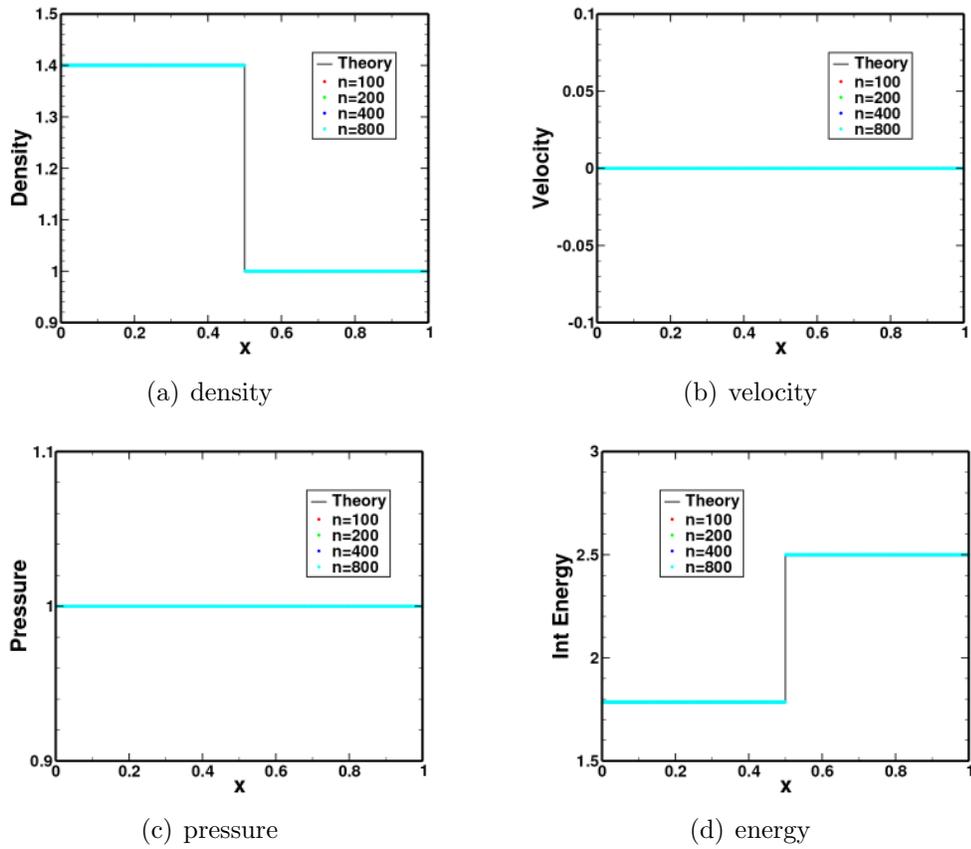


Figure 3.15: Single-phase numerical results for Toro Test #6. The tests used an *Eulerian grid* with the *predictor on*.

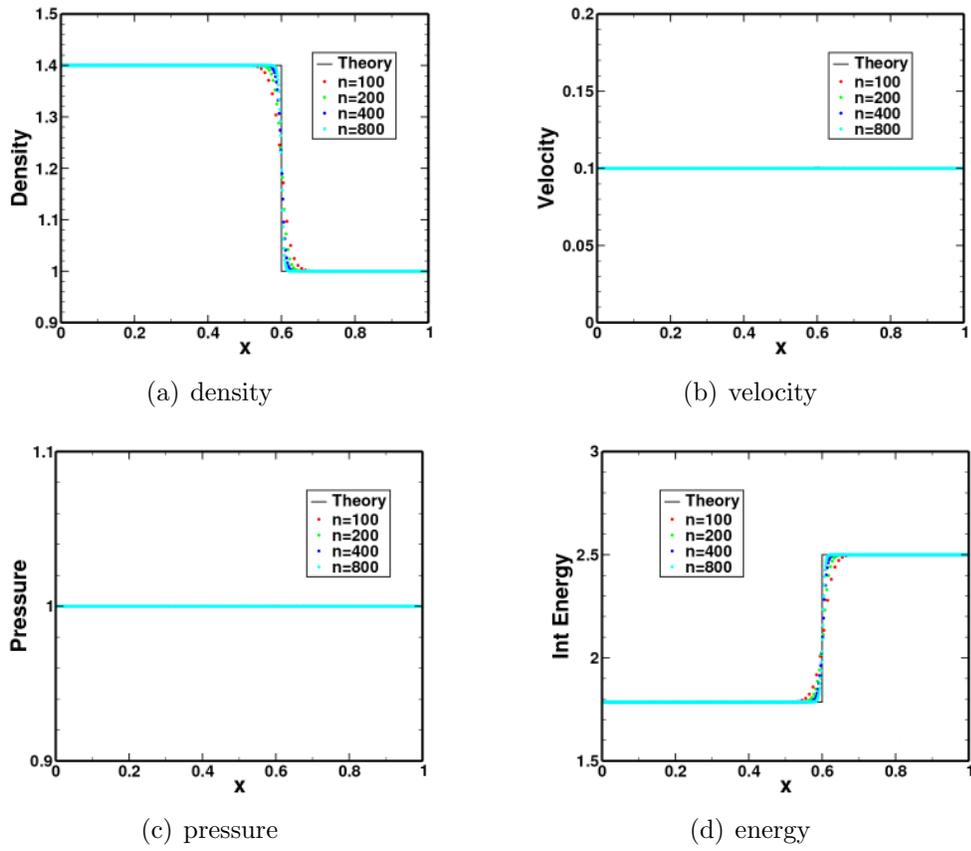


Figure 3.16: Single-phase numerical results for Toro Test #7. The tests used an *Eulerian grid* with the *predictor on*.

Test	x_0	t_0
1	0.3	0.2
2	0.5	0.15
3	0.5	0.012
4	0.4	0.035
5	0.8	0.012
6	0.5	2.0
7	0.4	2.0

Table 3.4: Position of discontinuity and end time for the Toro test problems.

Test	ρ_L	u_L	p_L	ρ_R	u_R	p_R
1	1.0	0.75	1.0	0.125	0.0	0.1
2	1.0	-2.0	0.4	1.0	2.0	0.4
3	1.0	0.0	1000.0	1.0	0.0	0.01
4	5.99924	19.5975	460.894	5.99242	-6.19633	46.0950
5	1.0	-19.59745	1000.0	1.0	-19.59745	0.01
6	1.4	0.0	1.0	1.0	0.0	1.0
7	1.4	0.1	1.0	1.0	0.1	1.0

Table 3.5: Initial conditions for the Toro test problems.

3.1.3 Sod Shock Tube

Test Description

The shock tube problem studied by Sod [80] has become a standard test for one-dimensional unsteady hydrodynamics algorithms. It consists of an ideal gas separated into two regions by a discontinuity in its state. Initially, the fluid is at rest and the left region is at a high pressure and the right region is at a lower pressure. The resulting flow-field consists of a rightward traveling shock, a rarefaction wave on the left, and in the middle there is a contact discontinuity following the shock.

The Sod test is a special case of the Toro test problems discussed in §3.1.2.

Test Setup

The Sod problem is a simulation of the single-phase Euler equations for an ideal gas with $\gamma = 1.4$. The tests consist of two constant states, $\{\rho_L, u_L, p_L\}$ on the left and $\{\rho_R, u_R, p_R\}$ on the right. The spatial domain of the problem is $x \in [0, 1]$ where the left and right states are separated by a discontinuity at position $x = 0.5$. The solutions are compared at the simulation end time $t = 0.15$. The initial conditions are tabulated in Table 3.6.

The problem is one dimensional by definition. It is modeled using a three-dimensional mesh with a single zone in the y and z directions and symmetry boundary conditions on the sides, resulting in a pseudo-1D flow in the x direction. The number of zones in the x direction are varied to look at mesh convergence. Outflow boundary conditions are given at the boundaries $x = 0$ and $x = 1$.

This test problem is run on both a fixed and moving mesh in order to compare the two methodologies. For the ALE Sod problem, the mesh motion is predefined and is not dependent on the motion of the material. The motion is defined such that the velocity of the center node (initially at $x = 0.5$) translates to the right at a constant speed. The center node's translation speed is specified such that it will travel one-quarter of the distance to the end of the domain at the end of the simulation.

Test Theory

The shock tube is a standard problem in gas dynamics. The theoretical solution to the flow in a shock tube is discussed in Anderson [82]. As the shock tube is a subset of the more general Riemann problem, the solution may also be computed using the Riemann solver of Toro [53].

Test Results

The numerical results of the Sod test are plotted in Figures 3.17–3.22. The numerical results for the various grid resolutions are given by the colored symbols, while the exact solution is represented by the solid black line. The computations were performed using both the first-order method (predictor off) and the second-order scheme (predictor on). Calculations were done using a fixed Eulerian mesh and a moving ALE mesh.

The results for the first-order scheme on the Eulerian mesh are shown in Figure 3.17. The numerical approximation exhibits some numerical diffusion. The shock has been smeared over 7 or 8 grid cells, and the contact is even more diffusive. The rarefaction seems to be reasonably resolved with only minor diffusion occurring at the head and tail of the wave. The calculation appears to be monotonic in the vicinity of the shock and contact as there are no spurious oscillations for this test case. The average position of the waves looks correct, indicating that speed of propagation of the waves is accurately captured in the calculations. The simulations appear to be converging to the theoretical solution as the mesh is refined.

The numerical results of the ALE Sod test are plotted in Figure 3.18. The results and convergence are comparable to the fixed-grid simulations. The fixed- and ALE-grid simulations are directly compared in Figure 3.19. Comparison of the fixed-grid simulations to the cases with mesh motion reveal that the moving grids better resolve the shock and contact surface. This is due to a number of factors. The first is better grid resolution because the mesh motion is causing the mesh to move into the shock region. Another reason is that since the mesh is moving in the

same direction as the material, there is less up-winding of the solution, resulting in less numerical diffusion.

A Lagrangian calculation, where the mesh moves with the material, is also plotted for comparison. Note that the current algorithms will only allow for Lagrangian mesh motion in one-dimensional problems. The Lagrangian solution exhibits an overshoot in the energy at the contact surface. This is the classical wall-heating problem typical to Lagrangian Godunov-type methods. Although this is a well recognized phenomenon, it is not well understood.

The above simulations were also performed with the predictor on. These results are given in Figures 3.20–3.22. As expected, the results show less diffusion around discontinuities than the first-order simulations. The solution is generally monotonic, or at least any oscillations are much smaller than the ones observed in §3.1.2. This is because the discontinuities in the Sod test are much less extreme.

Test Conclusions

The Sod test problem provides a mild test for analyzing the accuracy of a numerical method for common wave structures encountered in typical hydrodynamic problems. The following conclusions can be drawn from these calculations:

- The calculations show that the current code using the AUSF Riemann solver can be used to accurately represent the hydrodynamic phenomena of shocks, rarefactions, and contact surfaces.
- The first-order scheme is fairly diffusive near shocks and contact surfaces, as would be expected for first-order methods.
- The second-order scheme is less diffusive near shocks and contact surfaces.
- The calculations of the Sod problem exhibit monotonic behavior around shocks and contact surfaces.
- The calculations on moving meshes show similar behavior as those run on

fixed meshes. Thus, it appears that the ALE techniques used here work properly for this test.

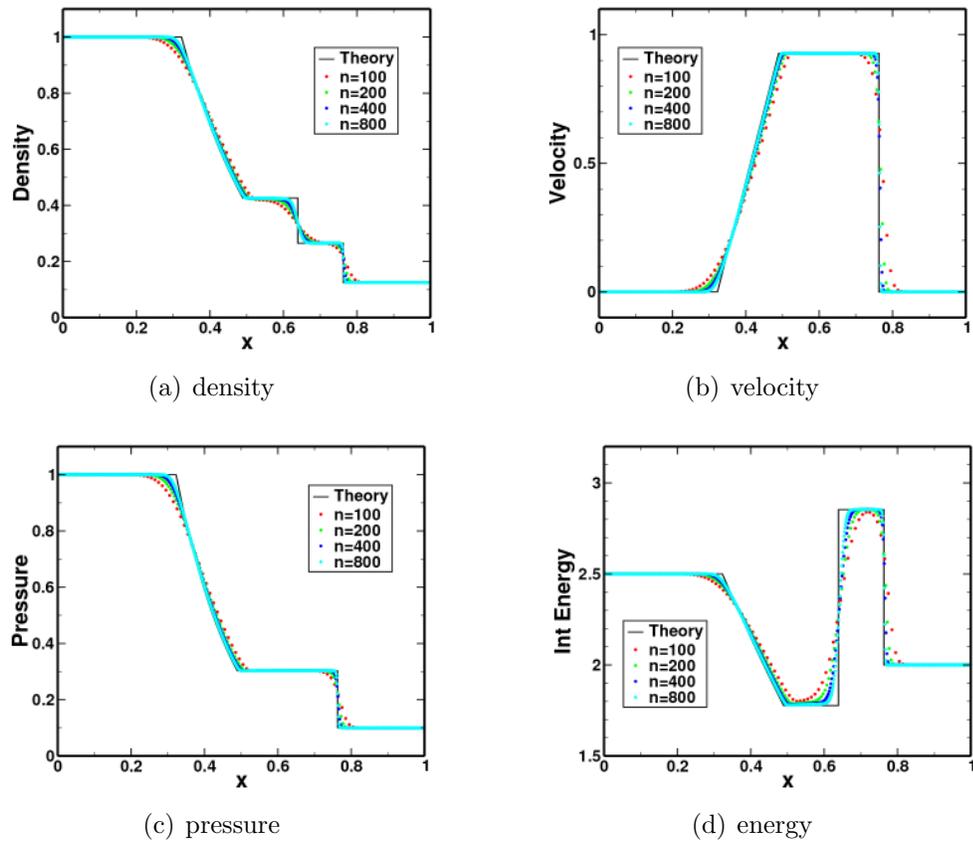


Figure 3.17: Single-phase numerical results for the Sod Test. The tests used an *Eulerian grid* with the *predictor off*.

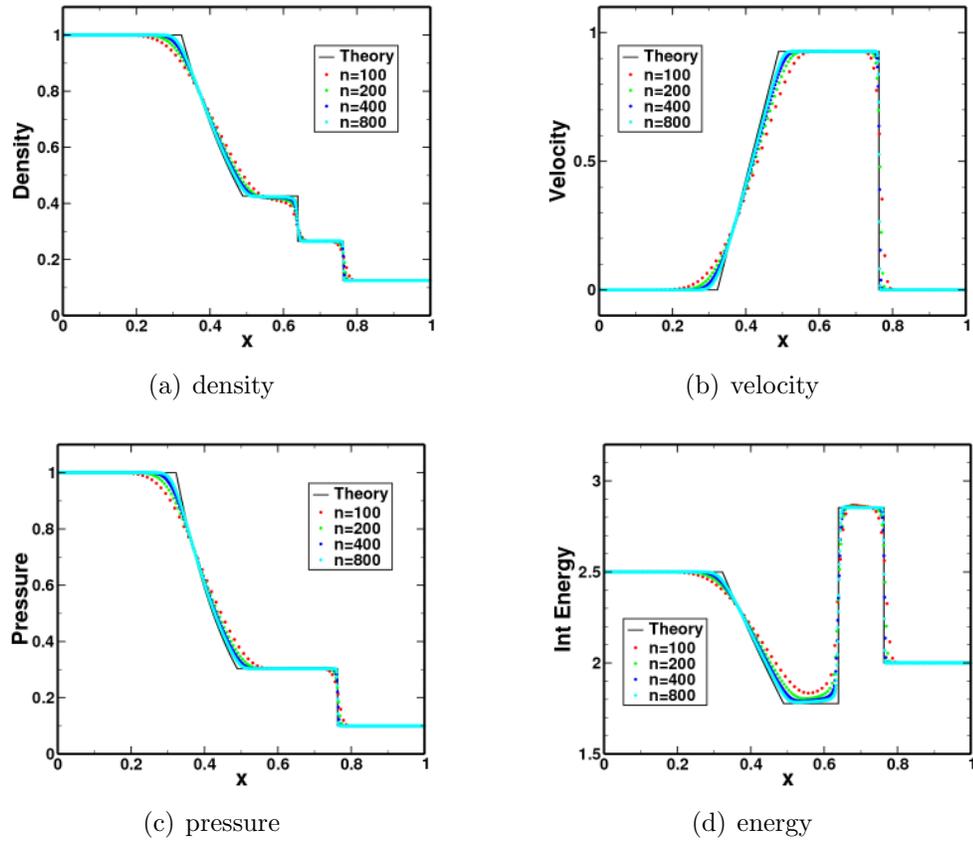


Figure 3.18: Single-phase numerical results for the Sod Test. The tests used an *ALE grid* with the *predictor off*.

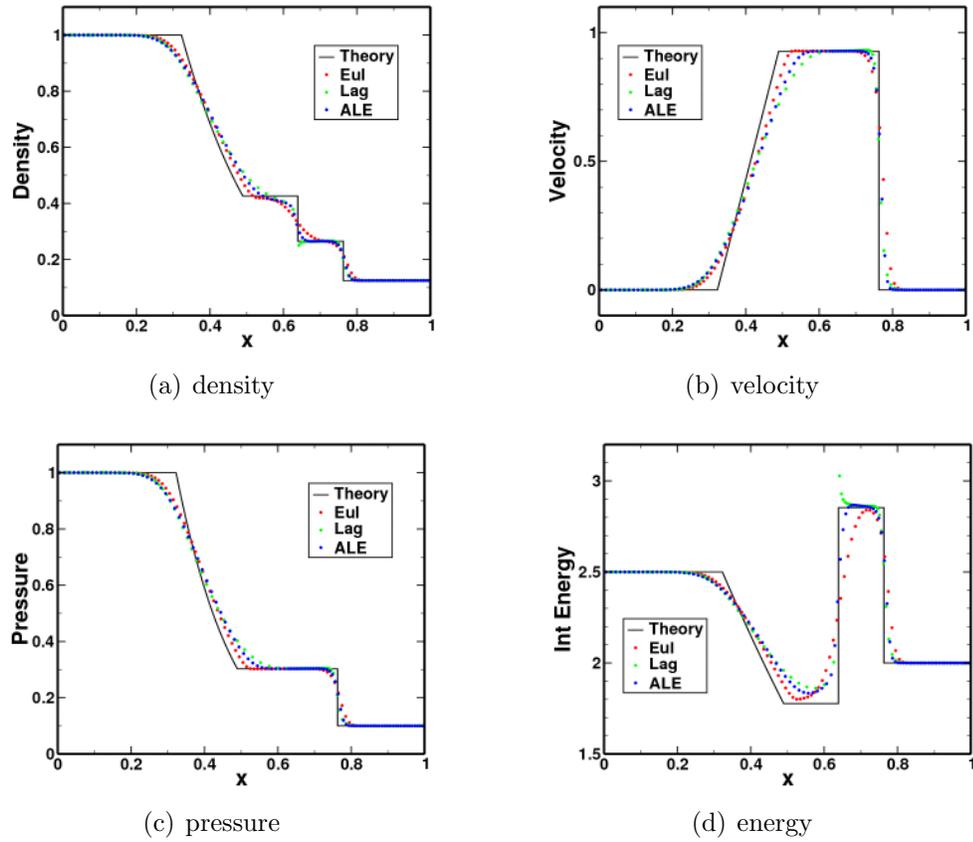


Figure 3.19: Single-phase numerical results for the Sod Test. The tests used *Eulerian/Lagrangian/ALE* grids with the *predictor off*.

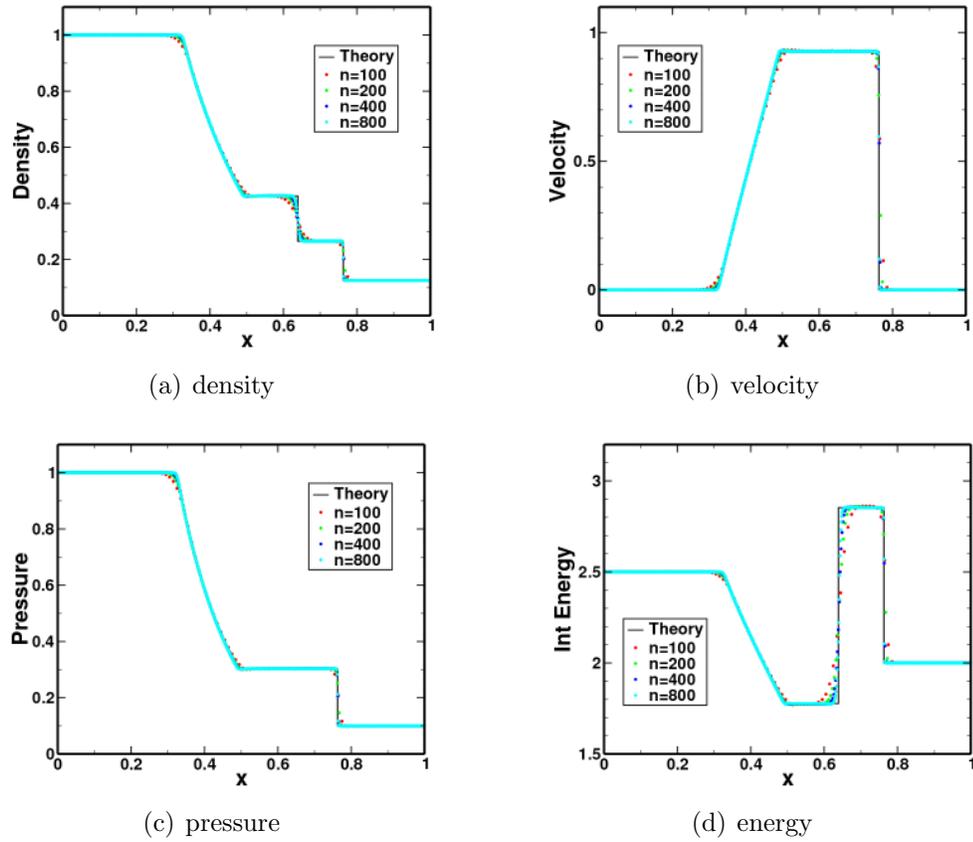


Figure 3.20: Single-phase numerical results for the Sod Test. The tests used an *Eulerian grid* with the *predictor on*.

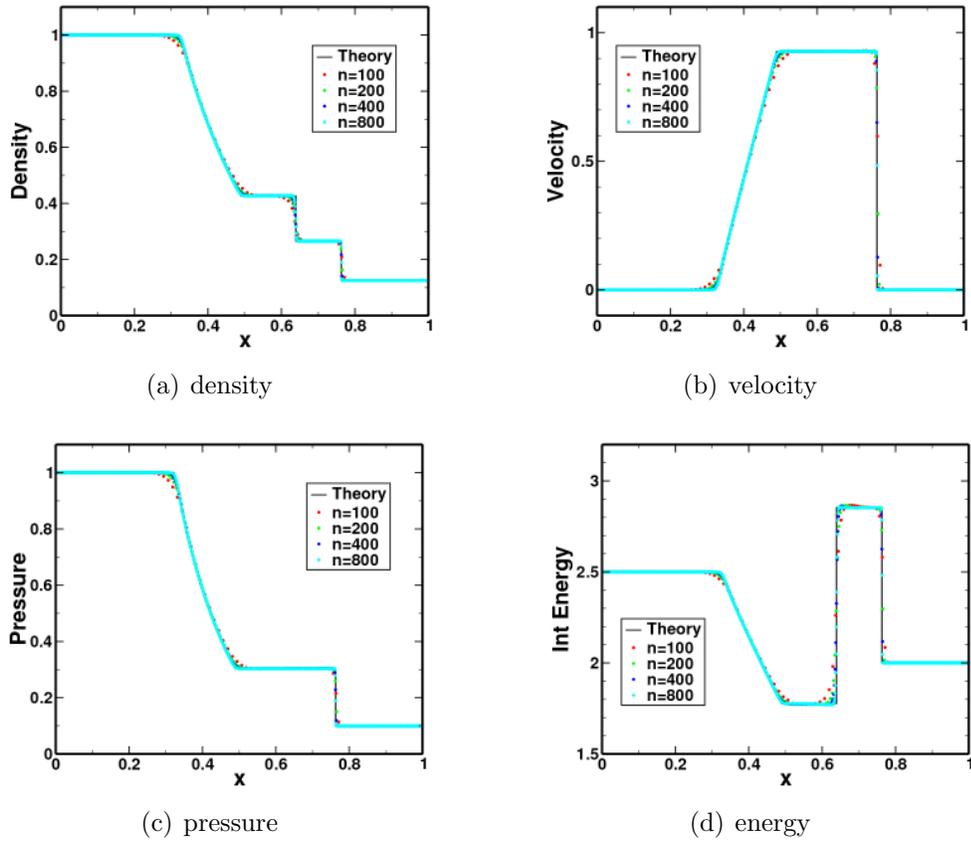


Figure 3.21: Single-phase numerical results for the Sod Test. The tests used an *ALE grid* with the *predictor on*.

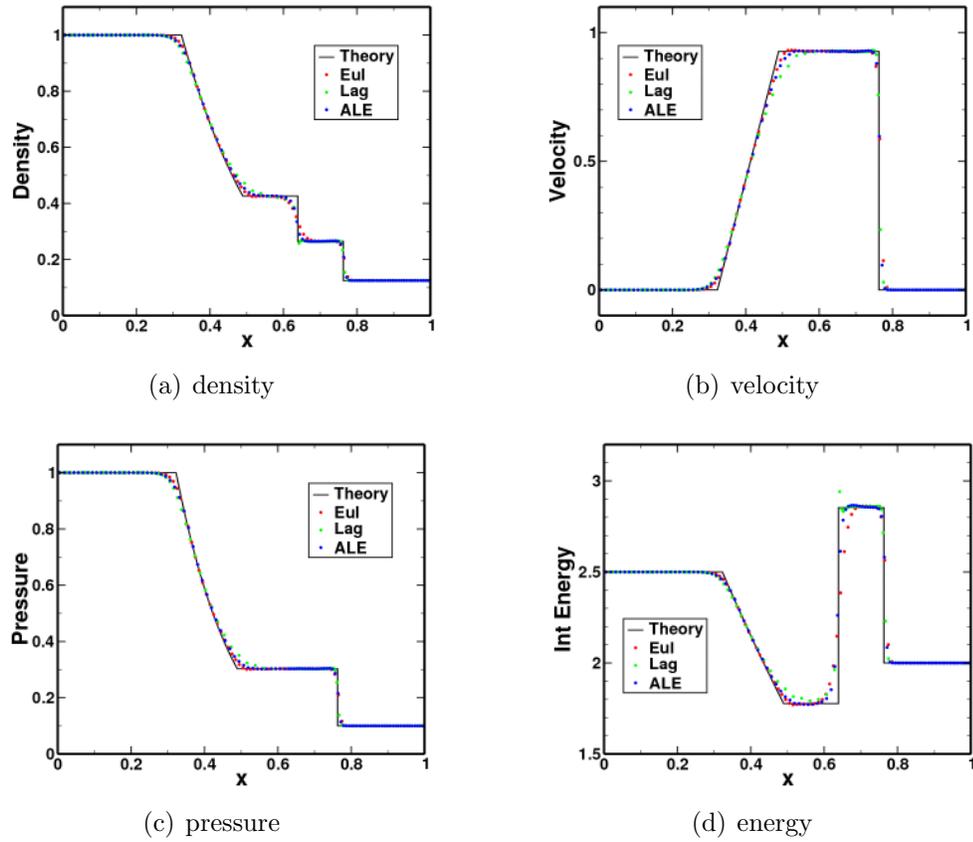


Figure 3.22: Single-phase numerical results for the Sod Test. The tests used *Eulerian/Lagrangian/ALE* grids with the *predictor on*.

Variable	Value
x_0	0.5
t_0	0.15
ρ_L	1.0
u_L	0.0
p_L	1.0
ρ_R	0.125
u_R	0.0
p_R	0.1

Table 3.6: Initial conditions for the Sod test problem.

3.1.4 Sedov Test

Test Description

The Sedov test is a one-dimensional explosion resulting from a finite quantity of energy placed within an otherwise homogeneous domain. The energy originates at a non-dimensional singularity point located at the origin. The background medium is an ideal gas at zero pressure. The energy release results in a strong shock propagating through the gas. The problem is symmetric, with spatial variation only in the radial dimension. The problem can be specified in Cartesian, cylindrical, or spherical coordinates, resulting in a planer, circular, or spherical shock respectively.

Test Setup

The background domain is filled with an ideal gas with $\gamma = 1.4$ and an initial density of $\rho_\infty = 1.0$. Rather than zero pressure as specified in the description, the gas is initialized with a very small pressure, $p_\infty = 10^{-14}$.

The theoretical solution to this problem assumes a singular energy source at the origin. However, in practice, the code is initialized with an extensive quantity of energy, E_{blast} , deposited within a small volume, V_{blast} . This small volume is taken as a single computational cell at the origin of the domain. Spreading the energy out over this small volume should have minimal effect on the solution at later times away from the origin. The intensive internal energy within this volume is computed as

$$\varepsilon = \frac{E_{blast}}{\rho_\infty V_{blast}}$$

where the blast energy depends on the geometry, such that

$$\text{Cartesian: } E_{blast} = 0.0673185$$

$$\text{Cylindrical: } E_{blast} = 0.311357$$

$$\text{Spherical: } E_{blast} = 0.851072$$

as given by [83] in order to obtain the desired profiles at the end time, $t = 1.0$.

The solution was computed on the seven different meshes shown in Figure 3.23. These meshes were chosen in order to compute the solution for all the coordinate

geometries (Cartesian, cylindrical, and spherical) as well as take advantage of various levels of symmetry inherent in the problem.

Test 1, Figure 3.23(a), uses a three-dimensional mesh to compute the 1D Cartesian Sedov test. It consists of a row of n elements in the x direction and a single zone in the y and z directions. Symmetry boundaries are on the y and z planes. A symmetry wall is located at $x = 0$ and an outflow boundary is on the other outer x surface.

Test 2, Figure 3.23(b), uses a three-dimensional mesh to compute the 1D cylindrical Sedov test. It consists of a row of n elements in the x direction and a single zone in the y and z directions. The elements are arranged in a wedge geometry with an angle of 4 degrees separating the y planes. Symmetry boundaries are on the z planes and angle-wall boundaries are on the y planes. A symmetry wall is located at $x = 0$ and an outflow boundary is on the other outer x surface. The origin, $x = 0$, is actually displaced slightly in the x direction to avoid a degenerate element.

Test 3, Figure 3.23(c), uses a three-dimensional mesh to compute the 1D spherical Sedov test. It consists of a row of n elements in the x direction and a single zone in the y and z directions. The elements are arranged in a pyramid geometry with an angle of 4 degrees separating the y planes and z planes. Angle-wall boundaries are on the y and z planes. A symmetry wall is located at $x = 0$ and an outflow boundary is on the other outer x surface. The origin, $x = 0$, is actually displaced slightly in the x direction to avoid a degenerate element.

Test 4, Figure 3.23(d), uses a two-dimensional axisymmetric mesh to compute the 1D cylindrical Sedov test. It consists of a column of n elements in the y direction and a single zone in the x direction. The elements are arranged in a Cartesian block. Symmetry boundaries are on the x planes. A symmetry wall is located at $y = 0$ and an outflow boundary is on the other outer y surface.

Test 5, Figure 3.23(e), uses a two-dimensional mesh to compute the 1D cylindrical Sedov test. It consists of a Cartesian block of elements, uniformly

spaced with n elements in the x and y directions. Symmetry boundaries are on the $x = 0$ and $y = 0$ planes and an outflow boundary is on the other outer surfaces. Data for this test is taken along a diagonal line characterized by the vector $\{1, 1\}$.

Test 6, Figure 3.23(f), uses a two-dimensional axisymmetric mesh to compute the 1D spherical Sedov test. It consists of a Cartesian block of elements, uniformly spaced with n elements in the x and y directions. Symmetry boundaries are on the $x = 0$ and $y = 0$ planes and an outflow boundary is on the other outer surfaces. Data for this test is taken along a diagonal line characterized by the vector $\{1, 1\}$.

Test 7, Figure 3.23(g), uses a three-dimensional mesh to compute the 1D spherical Sedov test. It consists of a Cartesian block of elements, uniformly spaced with n elements in the x , y , and z directions. Symmetry boundaries are on the $x = 0$, $y = 0$, and $z = 0$ planes and an outflow boundary is on the other outer surfaces. Data for this test is taken along a diagonal line characterized by the vector $\{1, 1, 1\}$.

Test Theory

The Sedov test has a self-similar analytical solution. Sedov [84] provided a general analytical closed-form solution to the problem. The theoretical solution was computed using methods discussed in [83].

Test Results

The computations of the seven tests were performed on a fixed Eulerian mesh using both the first-order method (predictor off) and the second-order scheme (predictor on). The results are plotted in Figures 3.24–3.30 for the first-order method, while second-order results are plotted in Figures 3.31–3.37. The numerical results for the various grid resolutions are given by the colored symbols, while the exact solution is represented by the solid black line. The tests were run using four mesh resolutions, $n = 30, 60, 120, 240$, where n is the number of grid cells in each coordinate direction. Thus for Test 7, when $n = 240$ there are a total of $240^3 = 13,824,000$ computational cells. Generally, it appears that the simulations are converging to the theoretical value as the mesh resolution improves.

The results for the first-order method are fairly consistent for each geometry. The peak values for density, pressure, energy and velocity are under-predicted using the coarse meshes, but the predictions appear to improve as the mesh is refined. Note that the internal energy goes to infinity at the origin. Therefore, the calculations are not expected to be accurate in that region. The first-order scheme diffuses the shock front over many grid cells. But overall, the position of the shock front is predicted accurately in all cases. It is interesting to note that the calculated values for Test 4 are very similar to the values obtained for Test 2, hence the axisymmetric formulation is working as expected.

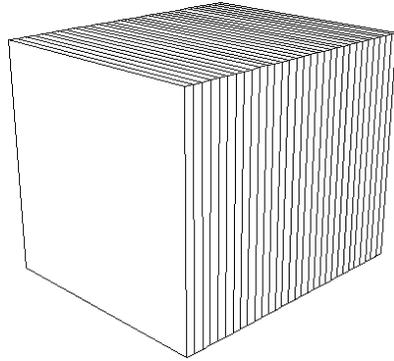
Turning the predictor on results in much less diffusive behavior. The coarse grids get much closer to the theoretical peak values and the shock is much sharper. The predictor does not appear to cause oscillations in the solution.

The calculations on multidimensional meshes seem to have some large scale fluctuations behind the shock. This is particularly evident in the velocity profiles (e.g Figure 3.37(b)). This behavior is most obvious when the predictor is on, but exists in the first-order calculations as well. This is due to angular variation in the solution and appears to be a consequence of the orientation of the vector along which the data was taken. The extent of the angular variation in the data can be seen in the scatter plots shown in Figures 3.38–3.43. Unlike the earlier plots, where data was taken along a specified line segment, the scatter plots gather data at every computational cell and plot it versus the radial position. These plots show that the shock position is predicted pretty consistently in all directions. The predicted gas velocity behind the shock, however, shows lots of scatter. Boundary effects could influence this behavior, although it is assumed that it is primarily caused by mesh imprinting. This is a result of the finite-volume method used, where shocks propagating across cell corners are not resolved as well as shocks propagating directly across the cell faces. This could be improved by including the influence of corner elements into the slope calculation used for the second-order predictor.

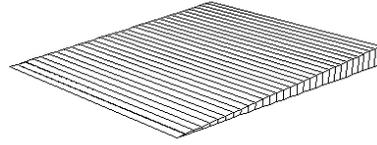
Test Conclusions

The following conclusions can be drawn from this calculation:

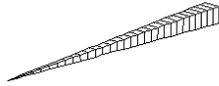
- The calculations show that the current code can be used to capture the blast wave associated with the Sedov test.
- The shock position is predicted well, although the peak values behind the shock are under-predicted. These values improve with additional mesh resolution.
- Scatter plots reveal that there is some mesh imprinting in the current results. This results in large angular variation in the solution.



(a) Test 1



(b) Test 2

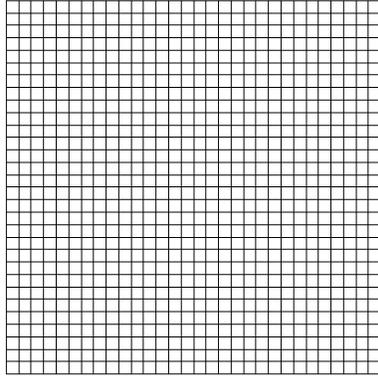


(c) Test 3

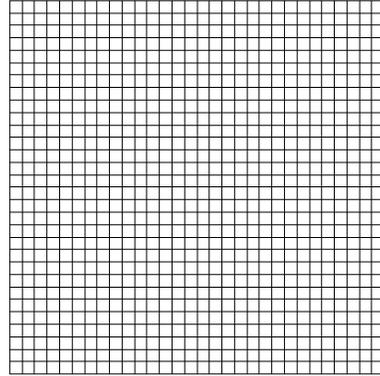


(d) Test 4

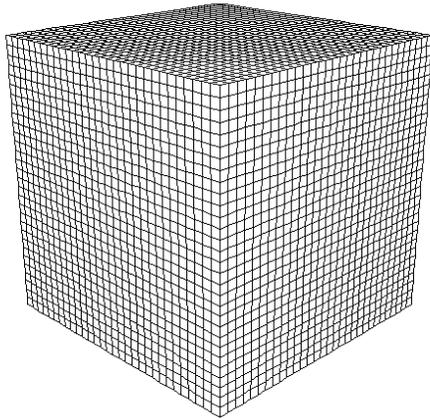
Figure 3.23: Initial mesh geometries for the Sedov tests.



(e) Test 5



(f) Test 6



(g) Test 7

Figure 3.23: Initial mesh geometries for the Sedov tests. (cont)

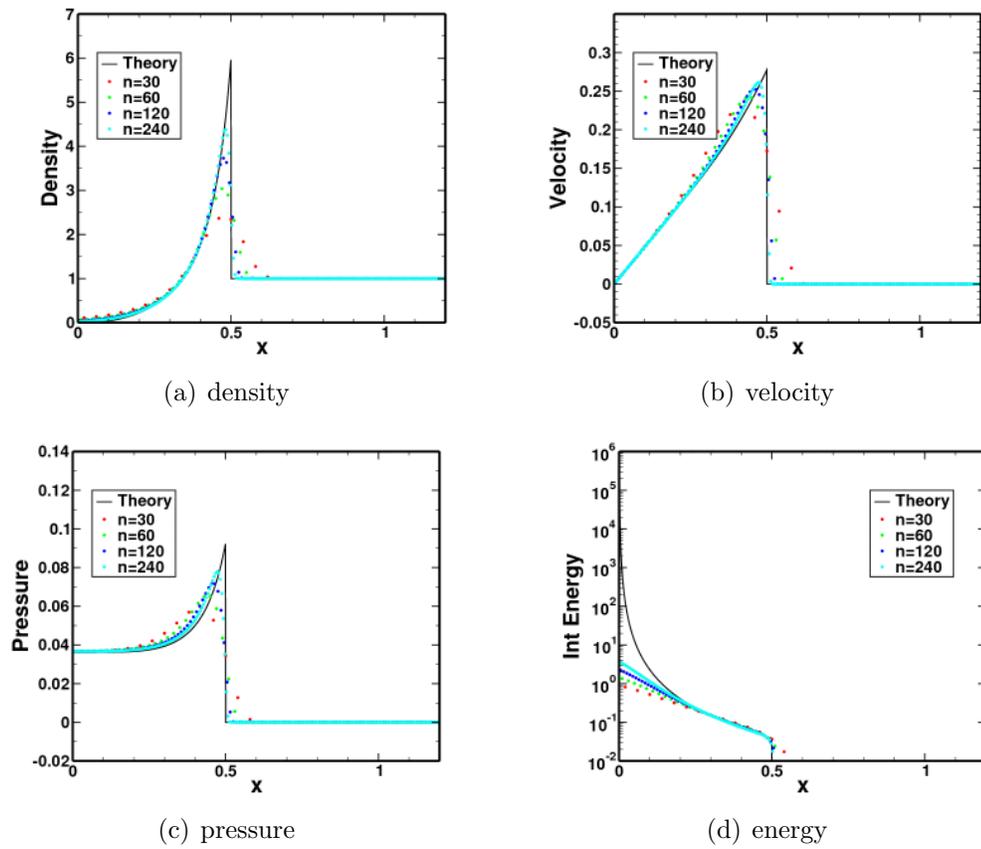


Figure 3.24: Single-phase numerical results for the Sedov Test #1 (1D Cartesian geometry). The tests used an *Eulerian grid* with the *predictor off*.

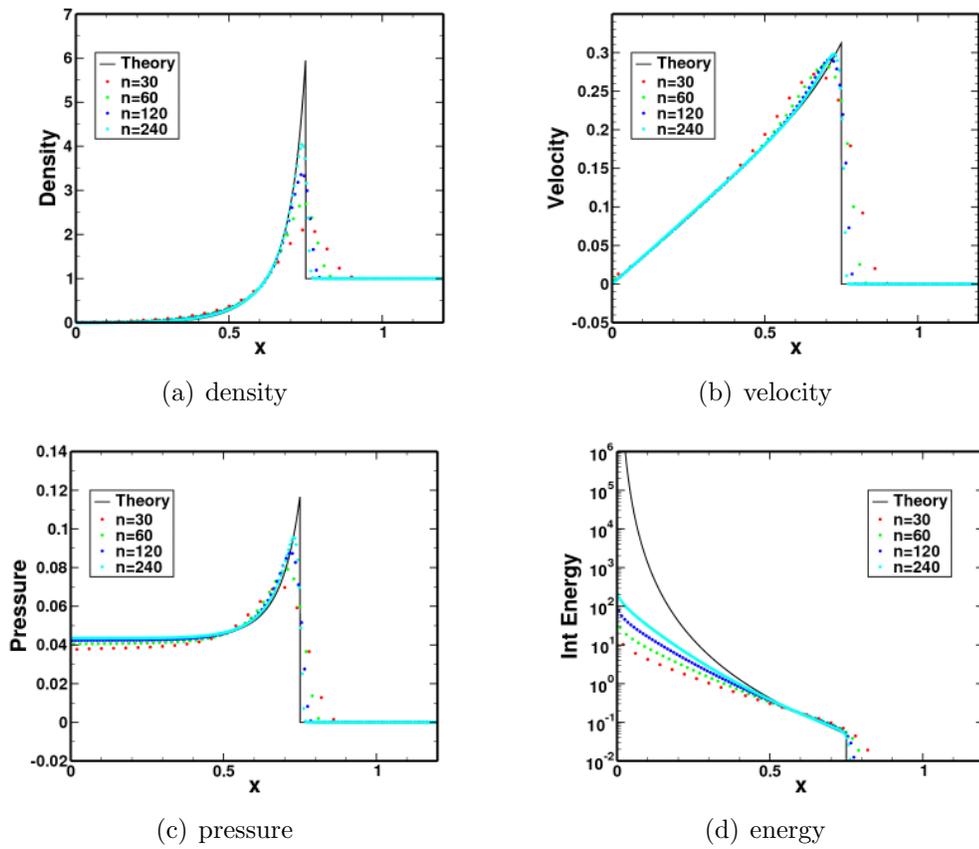


Figure 3.25: Single-phase numerical results for the Sedov Test #2 (1D Cylindrical geometry). The tests used an *Eulerian grid* with the *predictor off*.

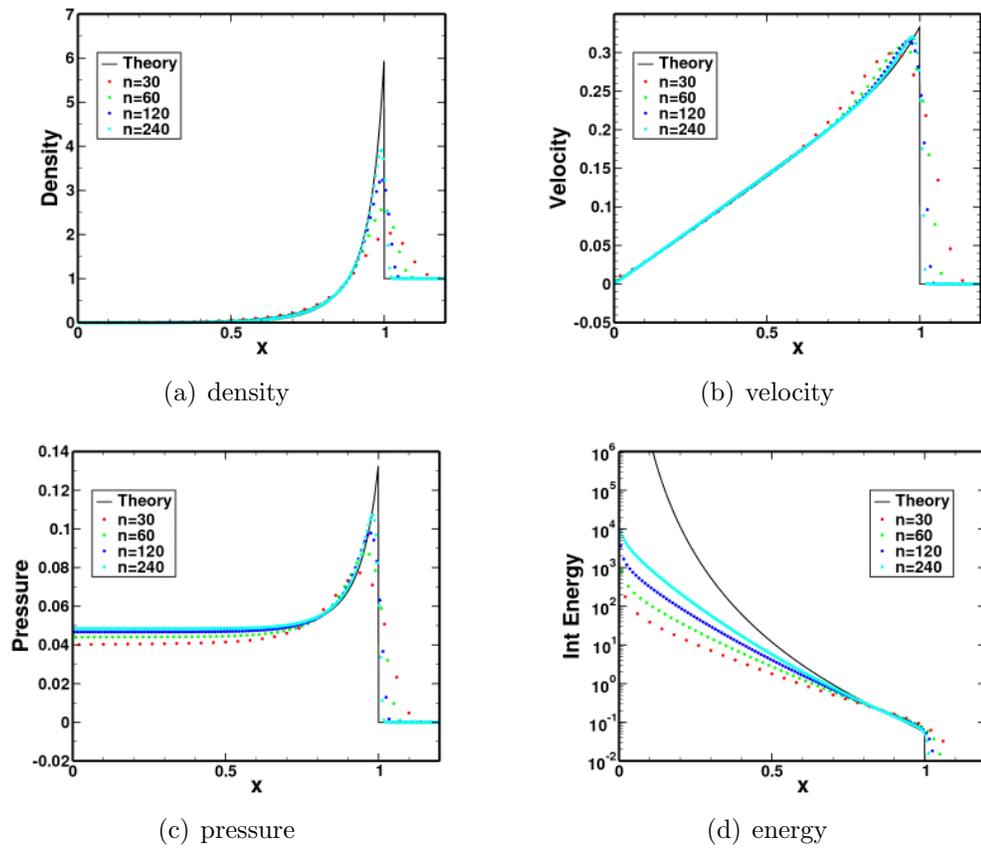


Figure 3.26: Single-phase numerical results for the Sedov Test #3 (1D Spherical geometry). The tests used an *Eulerian grid* with the *predictor off*.

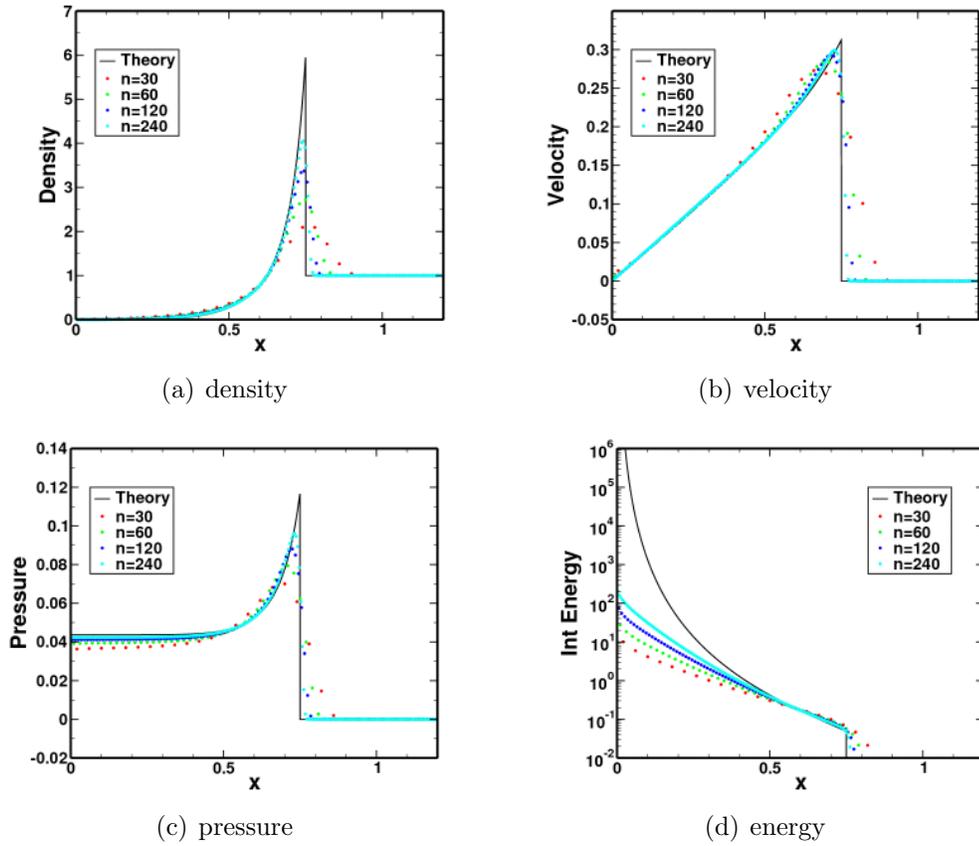


Figure 3.27: Single-phase numerical results for the Sedov Test #4 (1D-axisymmetric Cylindrical geometry). The tests used an *Eulerian grid* with the *predictor off*.

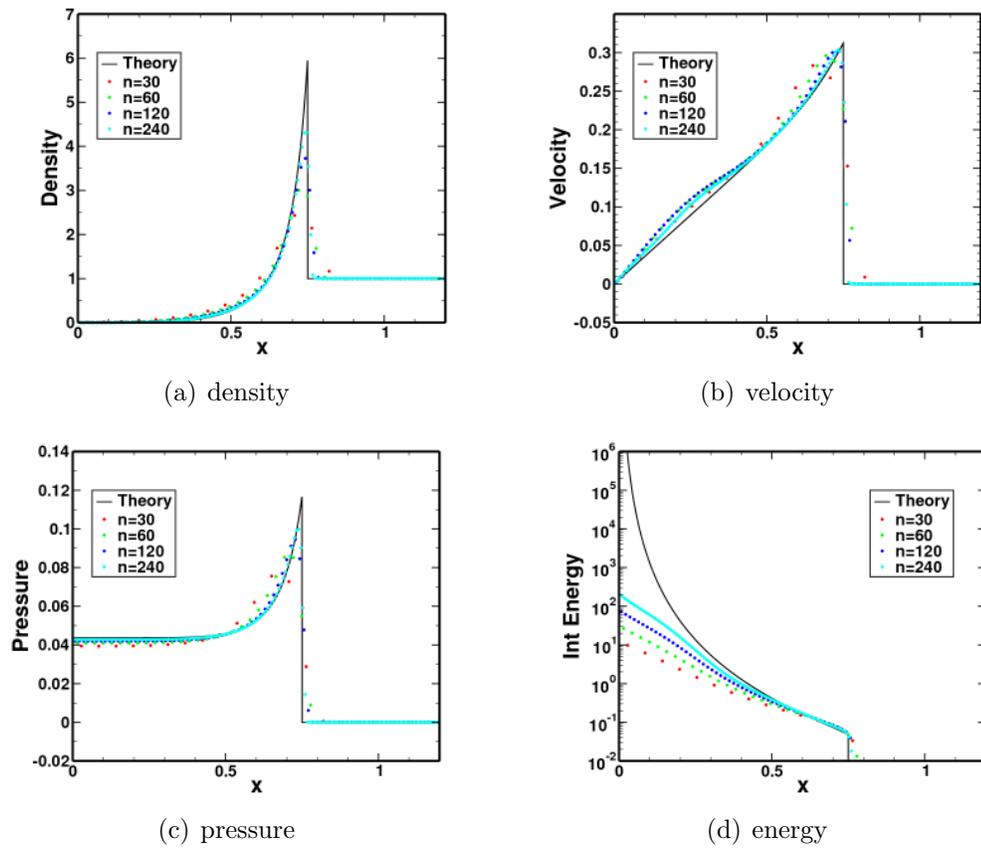


Figure 3.28: Single-phase numerical results for the Sedov Test #5 (2D Cylindrical geometry). The tests used an *Eulerian grid* with the *predictor off*.

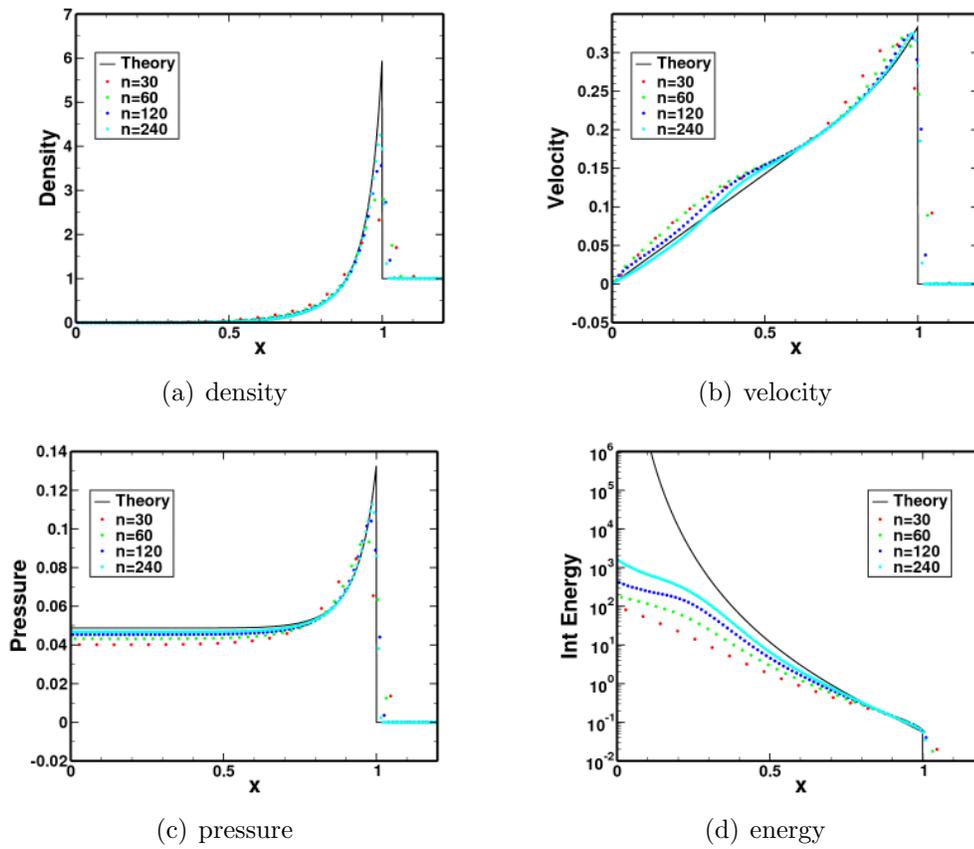


Figure 3.29: Single-phase numerical results for the Sedov Test #6 (2D-axisymmetric Spherical geometry). The tests used an *Eulerian grid* with the *predictor off*.

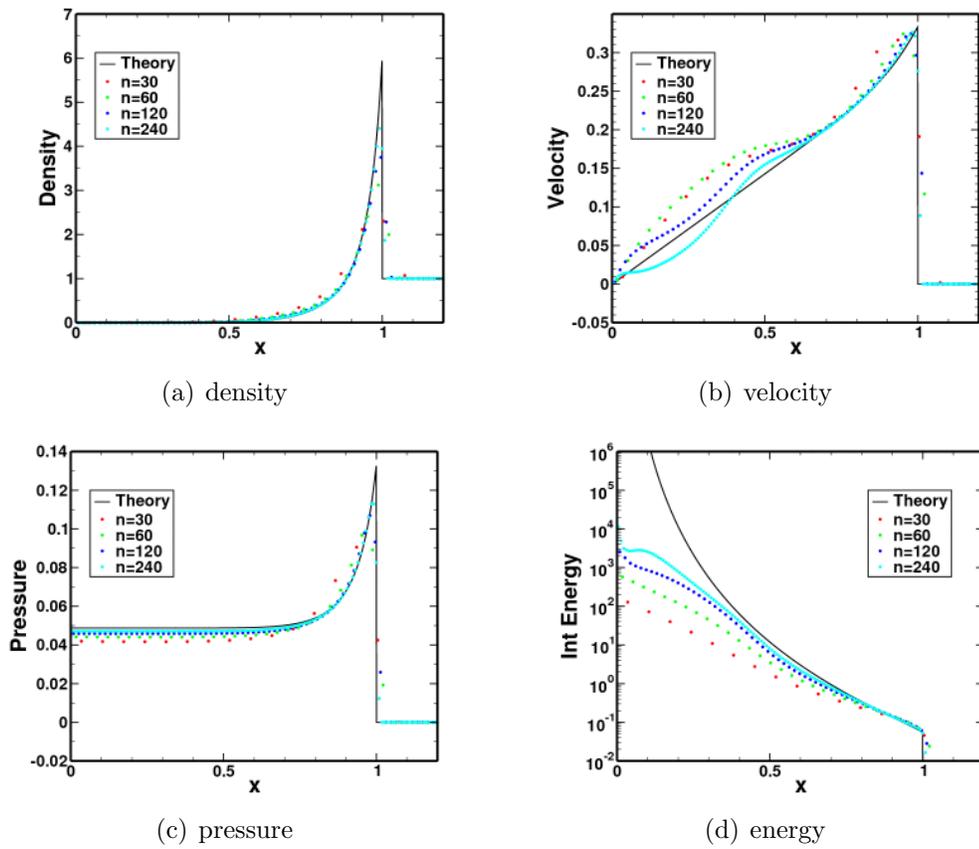


Figure 3.30: Single-phase numerical results for the Sedov Test #7 (3D Spherical geometry). The tests used an *Eulerian grid* with the *predictor off*.

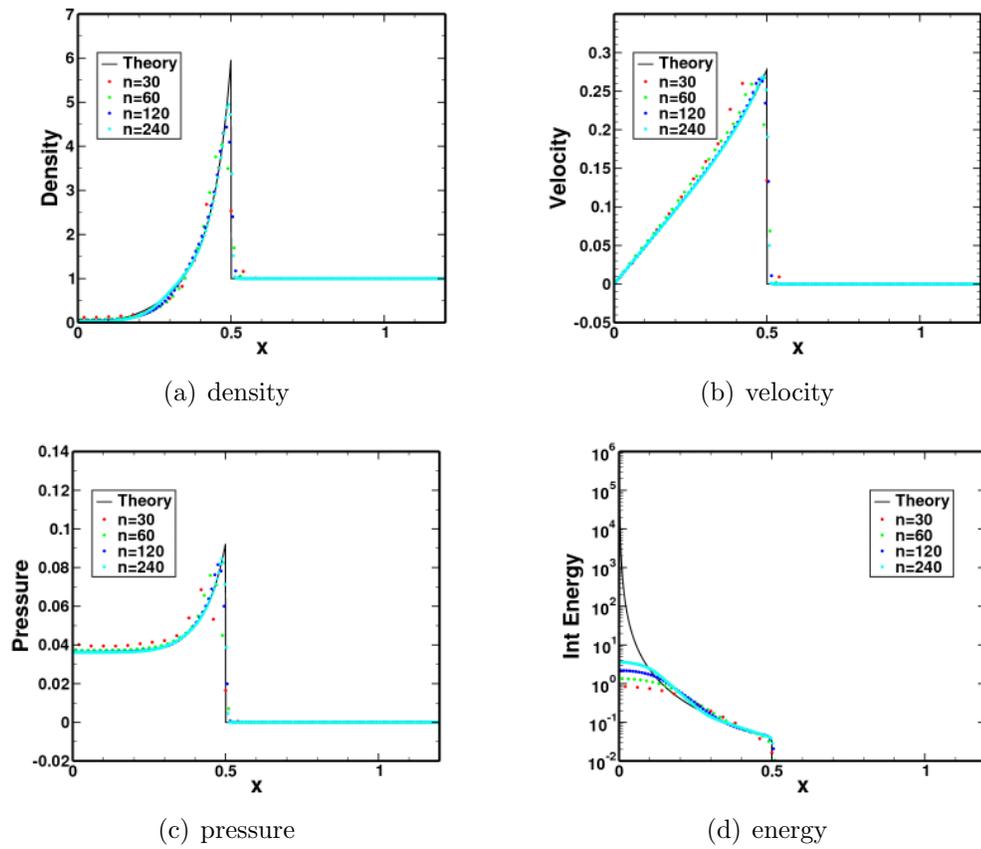


Figure 3.31: Single-phase numerical results for the Sedov Test #1 (1D Cartesian geometry). The tests used an *Eulerian grid* with the *predictor on*.

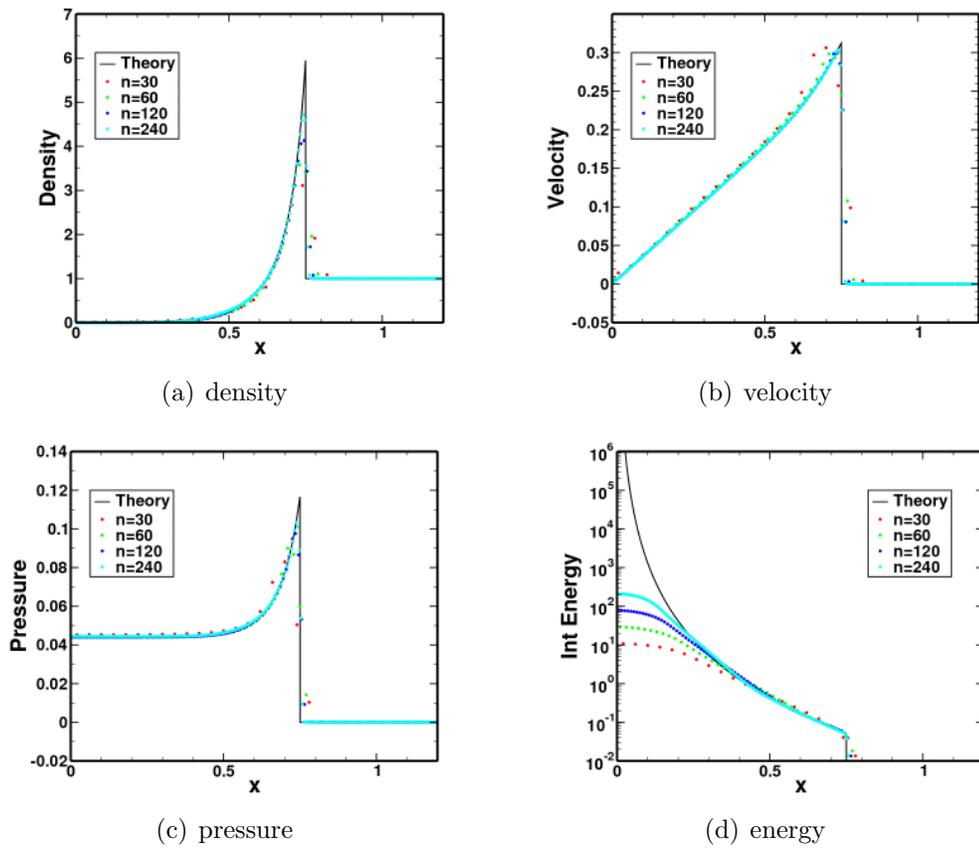


Figure 3.32: Single-phase numerical results for the Sedov Test #2 (1D Cylindrical geometry). The tests used an *Eulerian grid* with the *predictor on*.

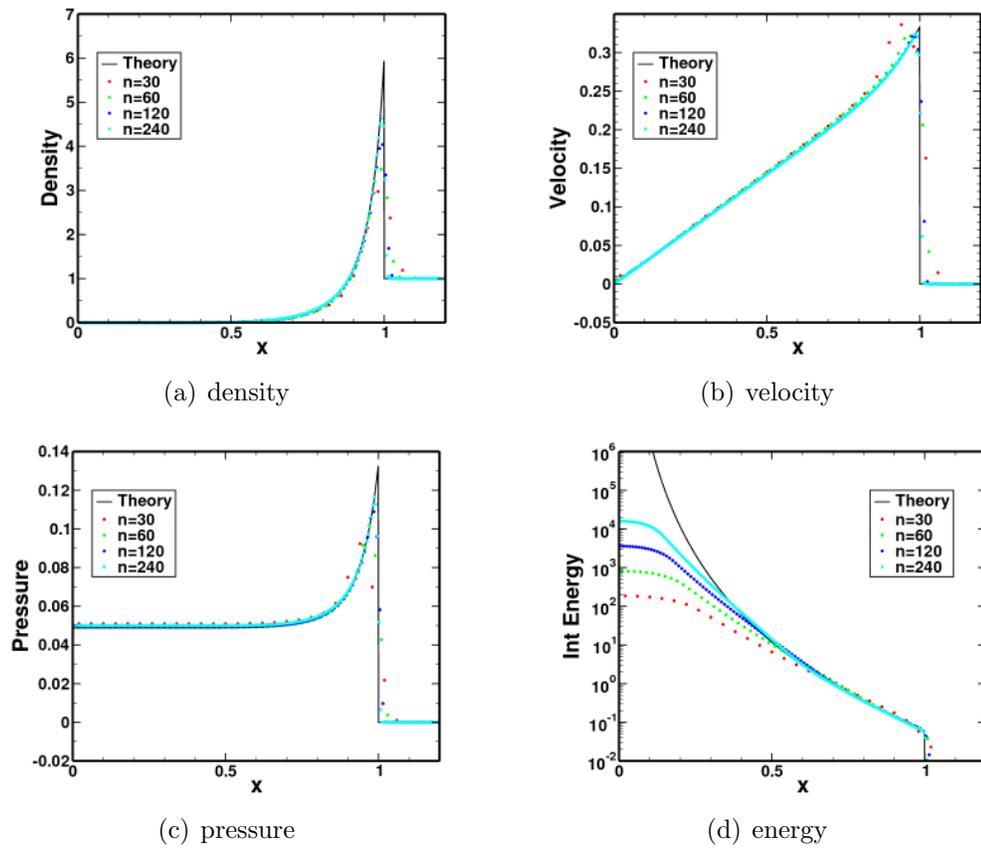


Figure 3.33: Single-phase numerical results for the Sedov Test #3 (1D Spherical geometry). The tests used an *Eulerian grid* with the *predictor on*.

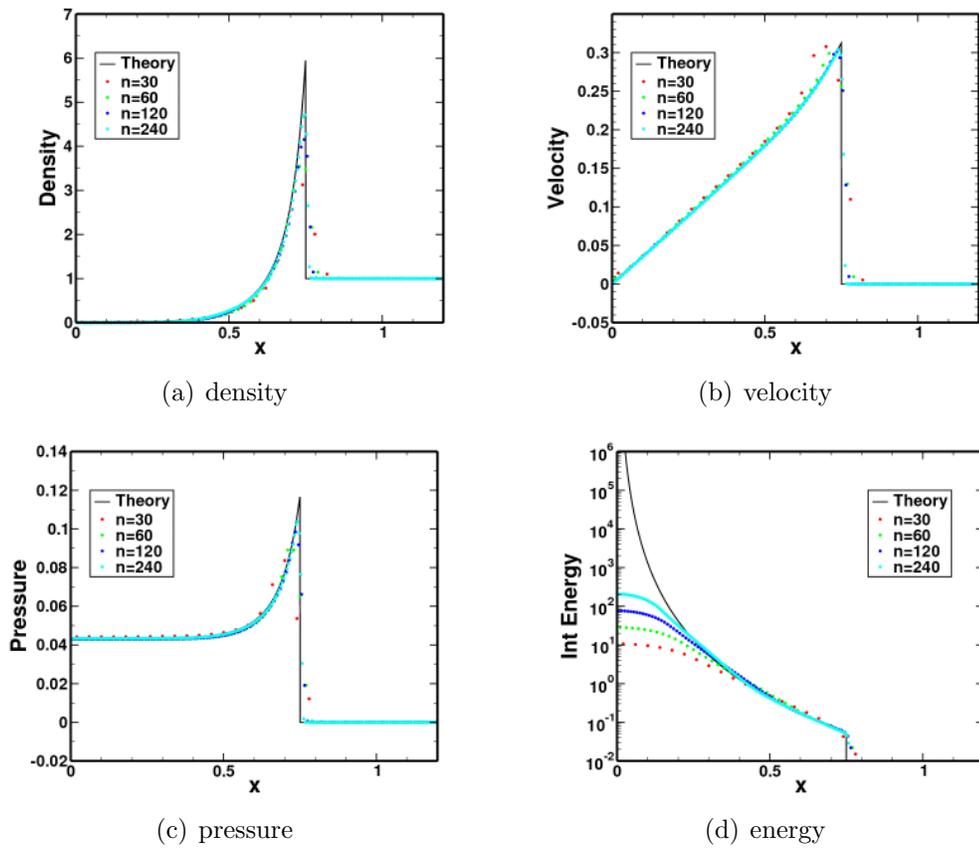


Figure 3.34: Single-phase numerical results for the Sedov Test #4 (1D-axisymmetric Cylindrical geometry). The tests used an *Eulerian grid* with the *predictor on*.

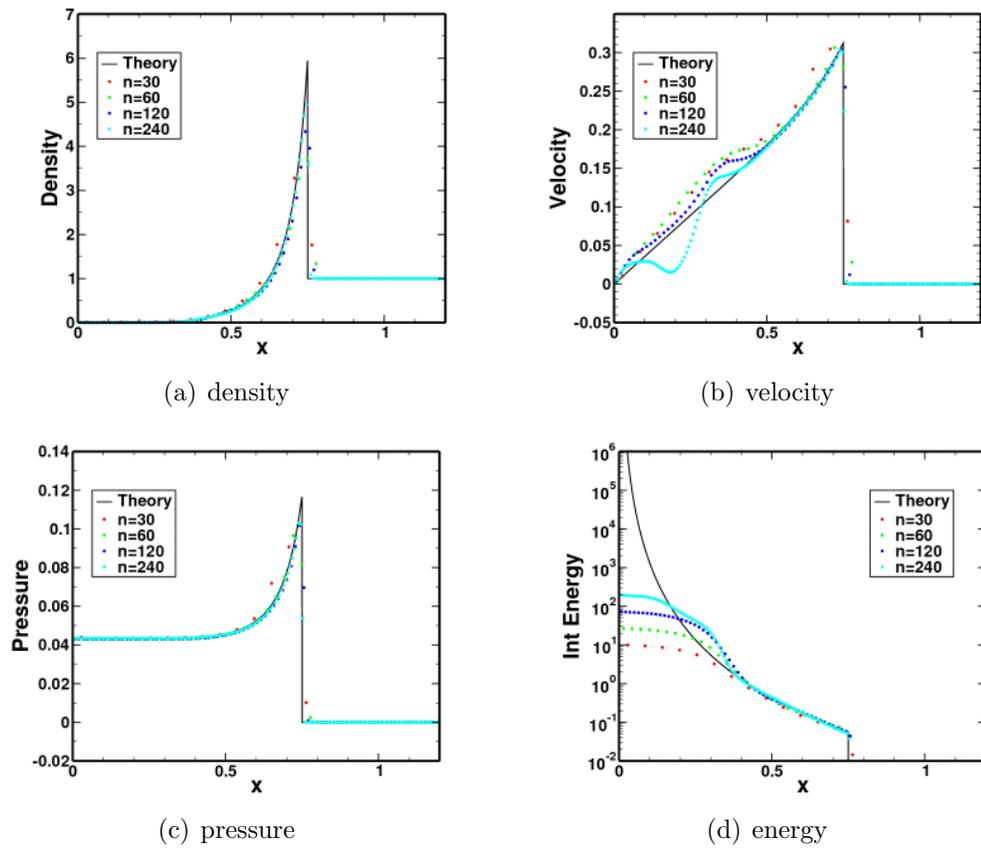


Figure 3.35: Single-phase numerical results for the Sedov Test #5 (2D Cylindrical geometry). The tests used an *Eulerian grid* with the *predictor on*.

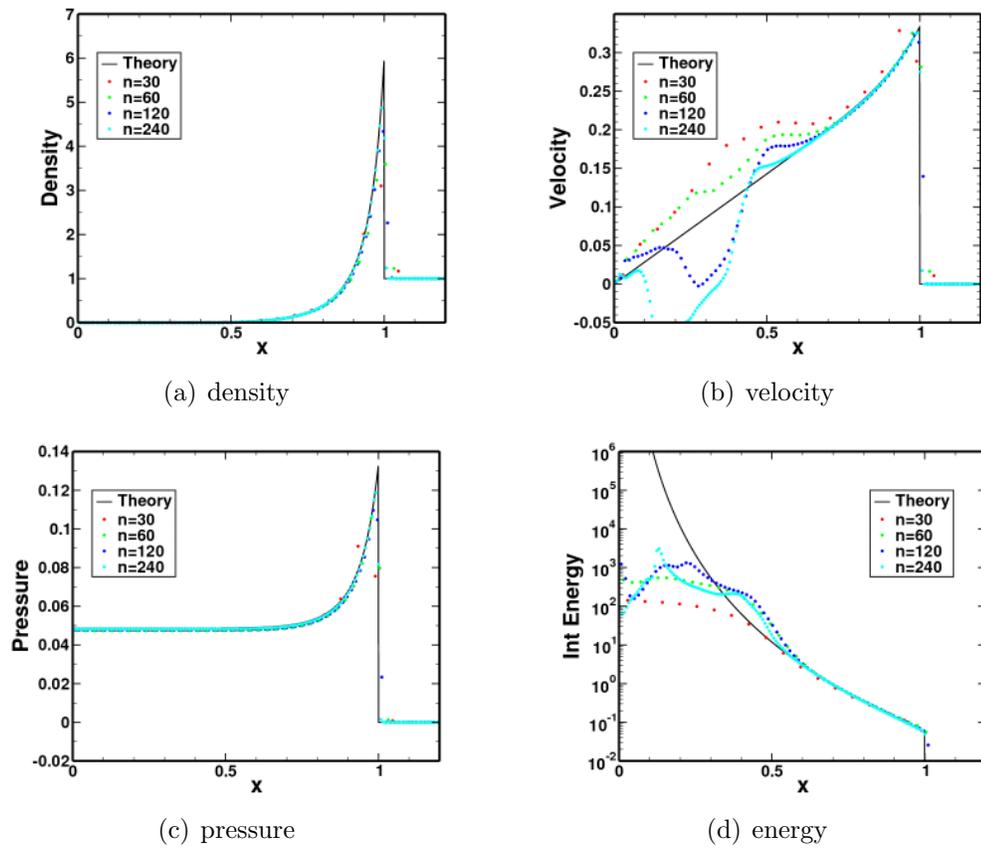


Figure 3.36: Single-phase numerical results for the Sedov Test #6 (2D-axisymmetric Spherical geometry). The tests used an *Eulerian grid* with the *predictor on*.

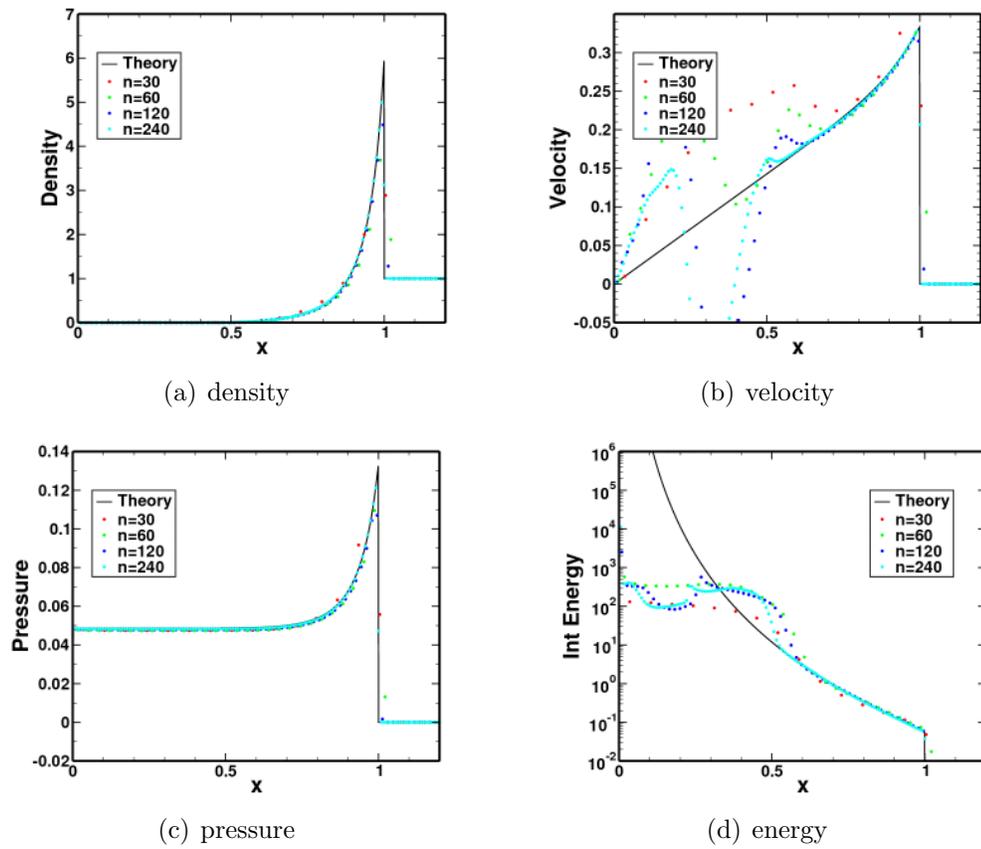


Figure 3.37: Single-phase numerical results for the Sedov Test #7 (3D Spherical geometry). The tests used an *Eulerian grid* with the *predictor on*.

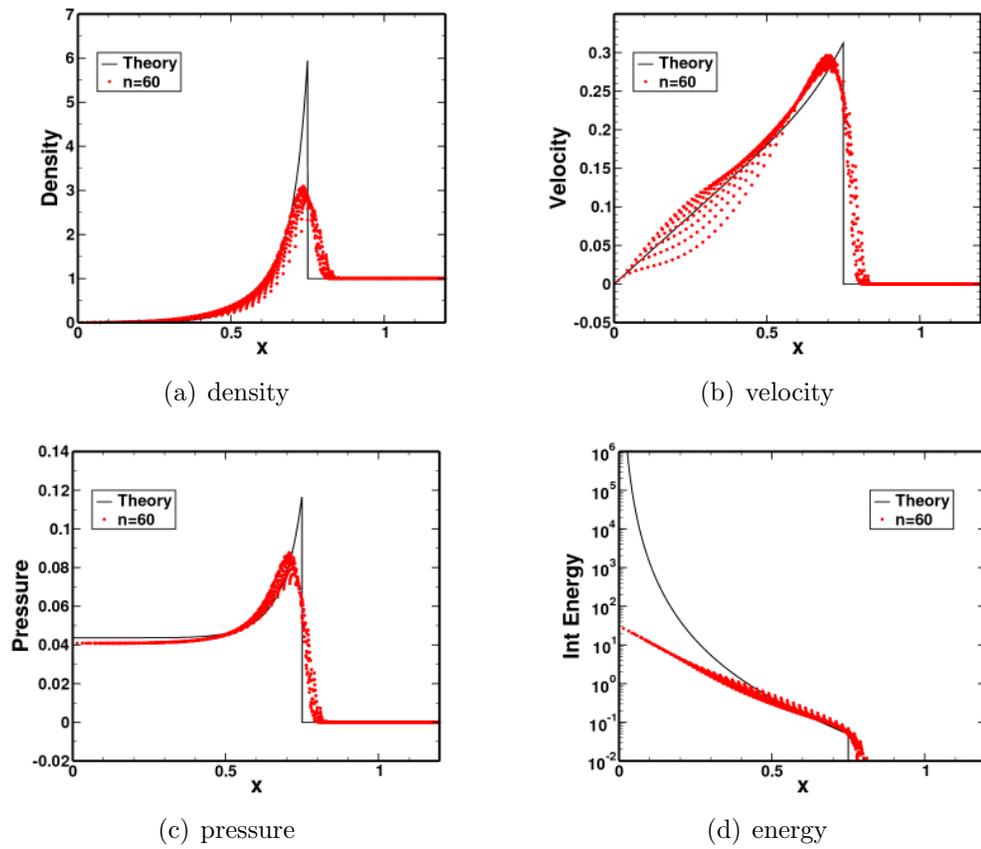


Figure 3.38: Single-phase numerical scatter-plot results for the Sedov Test #5 (2D Cylindrical geometry). The tests used an *Eulerian grid* with the *predictor off*.

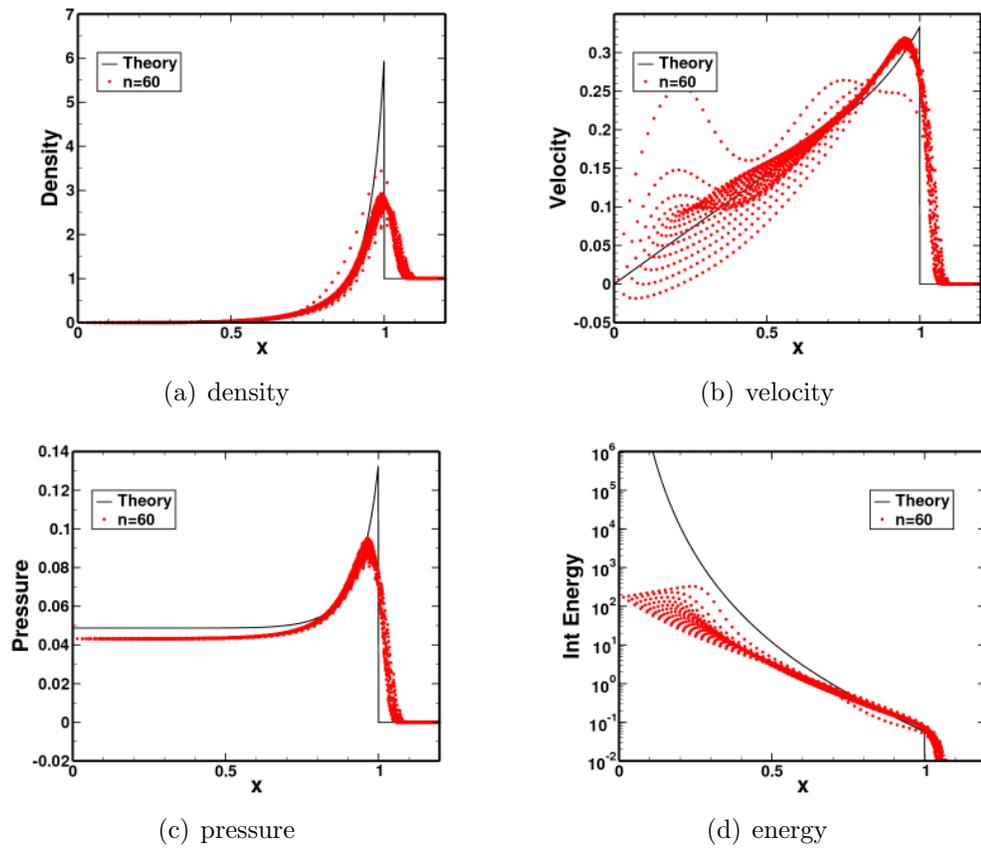


Figure 3.39: Single-phase numerical scatter-plot results for the Sedov Test #6 (2D-axisymmetric Spherical geometry). The tests used an *Eulerian grid* with the *predictor off*.

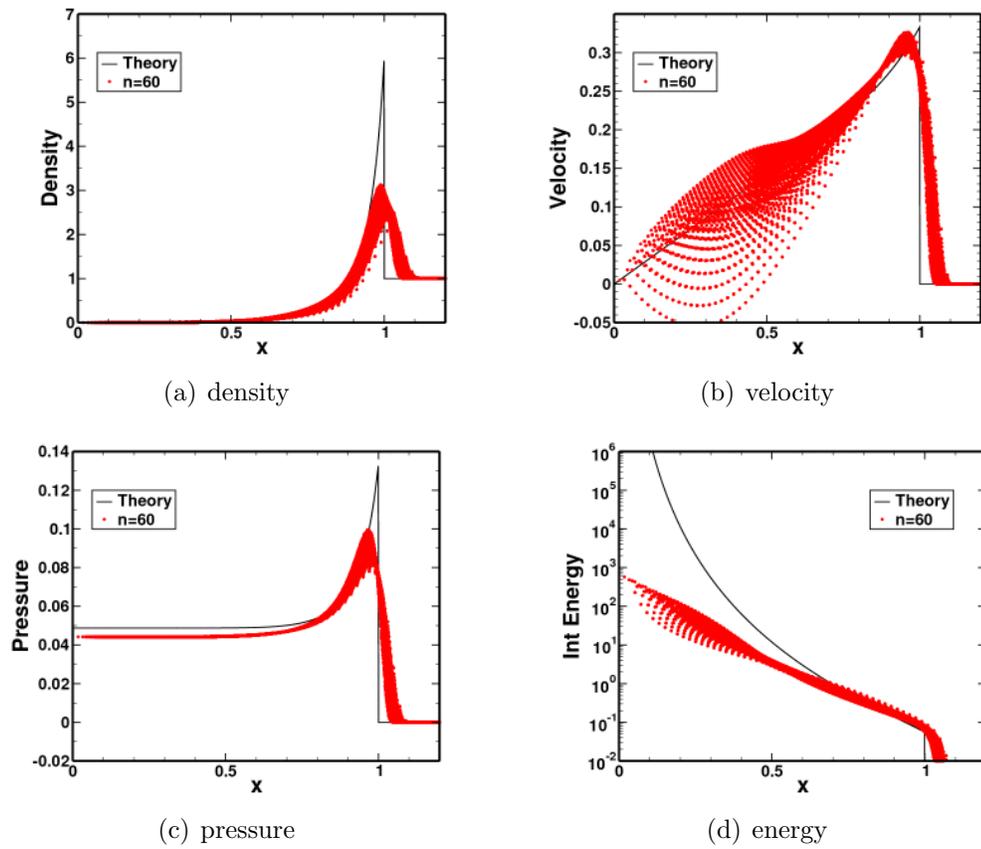


Figure 3.40: Single-phase numerical scatter-plot results for the Sedov Test #7 (3D Spherical geometry). The tests used an *Eulerian grid* with the *predictor off*.

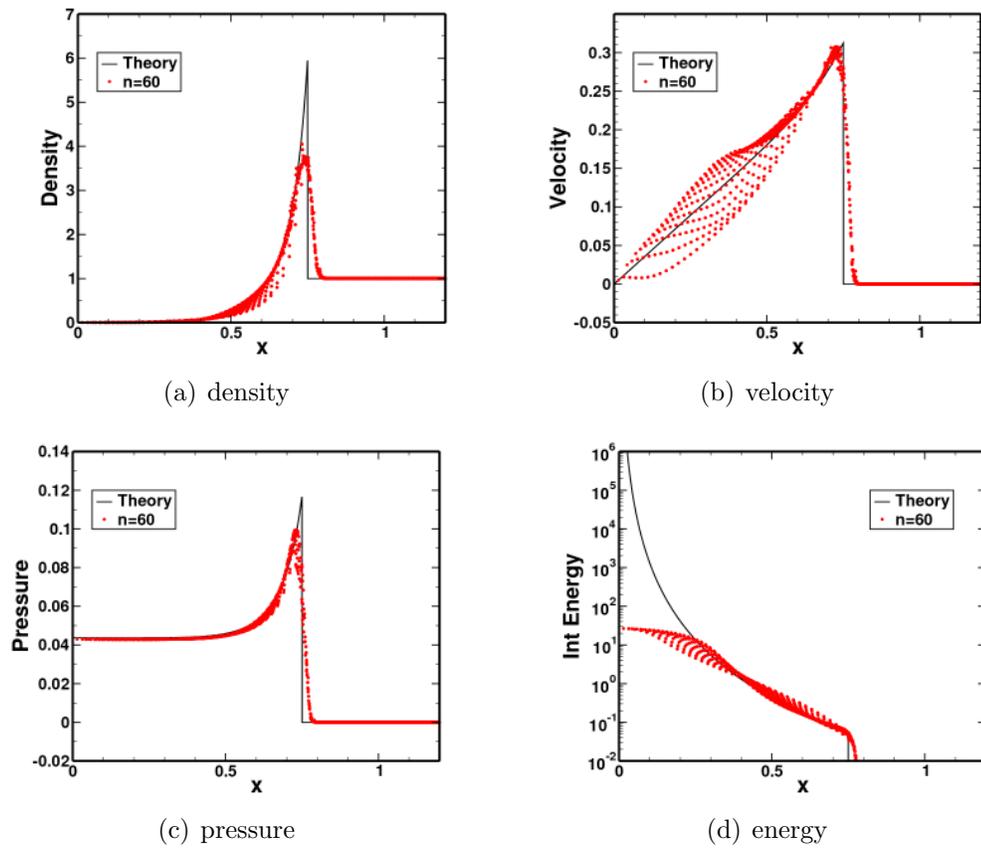


Figure 3.41: Single-phase numerical scatter-plot results for the Sedov Test #5 (2D Cylindrical geometry). The tests used an *Eulerian grid* with the *predictor on*.

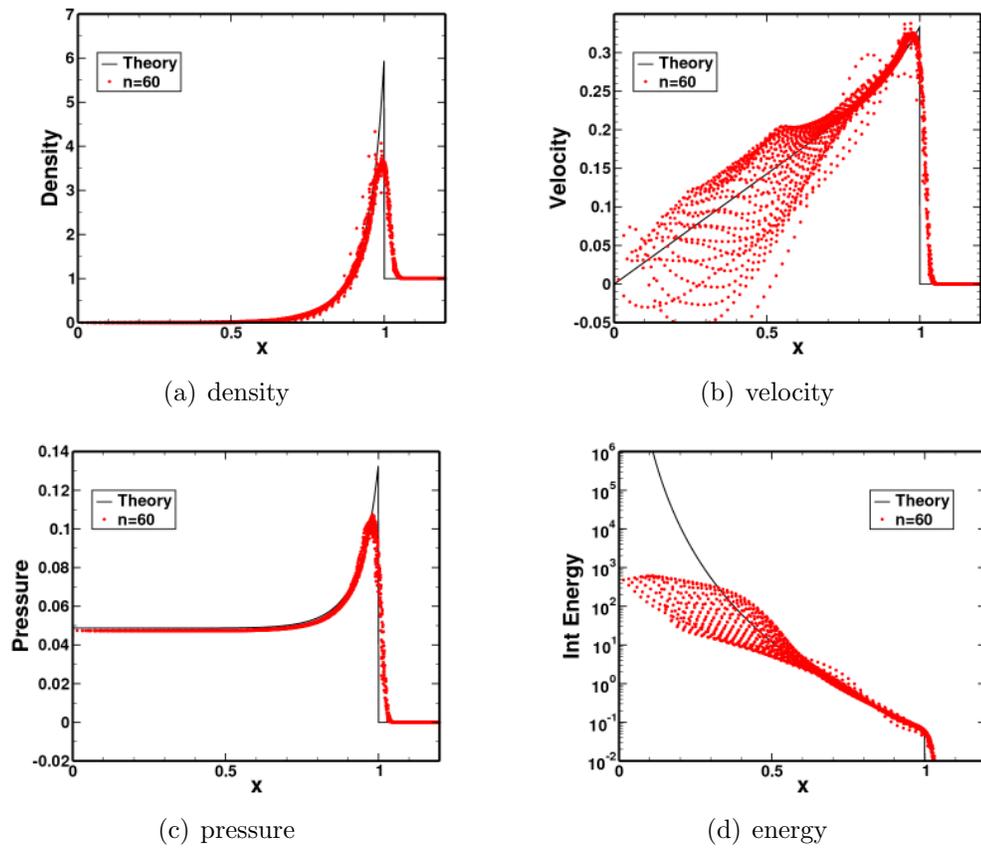


Figure 3.42: Single-phase numerical scatter-plot results for the Sedov Test #6 (2D-axisymmetric Spherical geometry). The tests used an *Eulerian grid* with the *predictor on*.

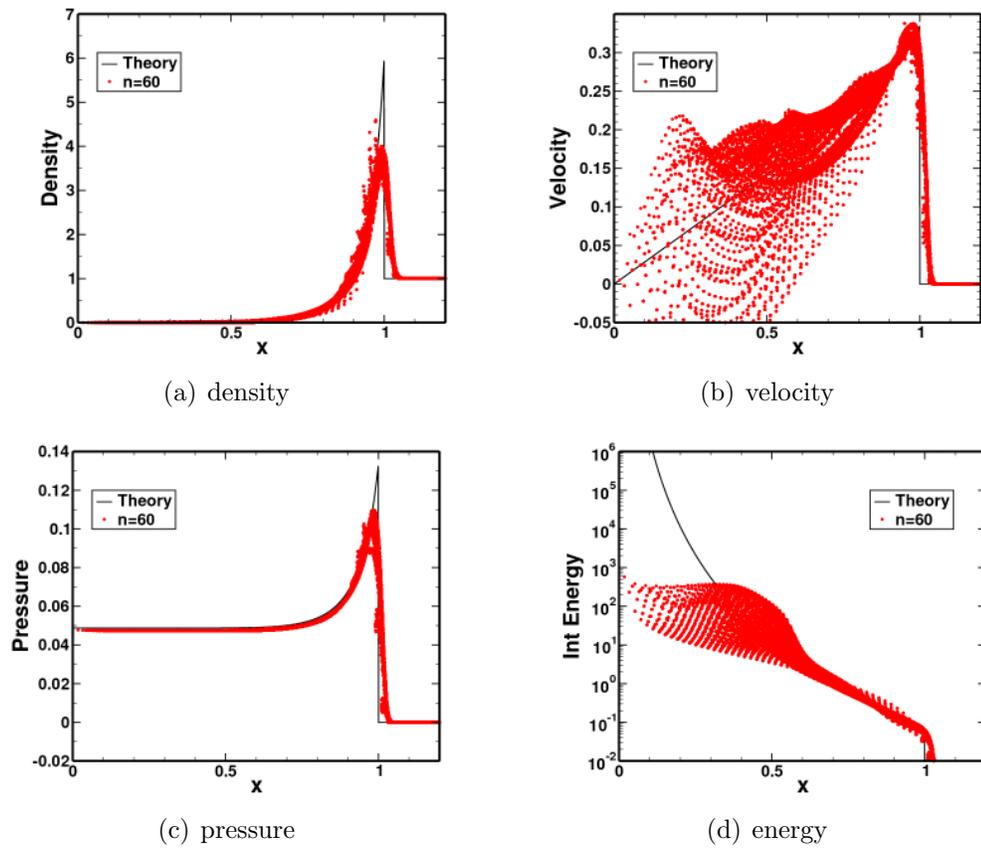


Figure 3.43: Single-phase numerical scatter-plot results for the Sedov Test #7 (3D Spherical geometry). The tests used an *Eulerian grid* with the *predictor on*.

3.1.5 Noh Test

Test Description

The Noh test is a one-dimensional gas implosion. The problem consists of a single material, an ideal gas, which is initialized with a uniform radially-inward velocity. This results in a shock at the origin which propagates outward as the gas is brought to rest. The problem is symmetric, with spatial variation only in the radial dimension. The problem can be specified in Cartesian, cylindrical, or spherical coordinates, resulting in a planer, circular, or spherical shock respectively.

Test Setup

The background domain is filled with an ideal gas with $\gamma = 5/3$ and an initial density of $\rho_0 = 1.0$. The velocity is initialized with the flow going inward in the radial direction such that the radial velocity is $v_0 = -1.0$.

The solution was computed on the seven different meshes shown in Figure 3.44. These meshes were chosen in order to compute the solution for all the coordinate geometries (Cartesian, cylindrical, and spherical) as well as take advantage of various levels of symmetry inherent in the problem.

Test 1, Figure 3.44(a), uses a three-dimensional mesh to compute the 1D Cartesian Noh test. It consists of a row of n elements in the x direction and a single zone in the y and z directions. Symmetry boundaries are on the y and z planes. A symmetry wall is located at $x = 0$ and an outflow boundary is on the other outer x surface.

Test 2, Figure 3.44(b), uses a three-dimensional mesh to compute the 1D cylindrical Noh test. It consists of a row of n elements in the x direction and a single zone in the y and z directions. The elements are arranged in a wedge geometry with an angle of 4 degrees separating the y planes. Symmetry boundaries are on the z planes and angle-wall boundaries are on the y planes. A symmetry wall is located at $x = 0$ and an outflow boundary is on the other outer x surface. The origin, $x = 0$, is actually displaced slightly in the x direction to avoid a degenerate element.

Test 3, Figure 3.44(c), uses a three-dimensional mesh to compute the 1D spherical Noh test. It consists of a row of n elements in the x direction and a single zone in the y and z directions. The elements are arranged in a pyramid geometry with an angle of 4 degrees separating the y planes and z planes. Angle-wall boundaries are on the y and z planes. A symmetry wall is located at $x = 0$ and an outflow boundary is on the other outer x surface. The origin, $x = 0$, is actually displaced slightly in the x direction to avoid a degenerate element.

Test 4, Figure 3.44(d), uses a two-dimensional axisymmetric mesh to compute the 1D cylindrical Noh test. It consists of a column of n elements in the y direction and a single zone in the x direction. The elements are arranged in a Cartesian block. Symmetry boundaries are on the x planes. A symmetry wall is located at $y = 0$ and an outflow boundary is on the other outer y surface.

Test 5, Figure 3.44(e), uses a two-dimensional mesh to compute the 1D cylindrical Noh test. It consists of a Cartesian block of elements, uniformly spaced with n elements in the x and y directions. Symmetry boundaries are on the $x = 0$ and $y = 0$ planes and an outflow boundary is on the other outer surfaces. Data for this test is taken along a diagonal line characterized by the vector $\{1, 1\}$.

Test 6, Figure 3.44(f), uses a two-dimensional axisymmetric mesh to compute the 1D spherical Noh test. It consists of a Cartesian block of elements, uniformly spaced with n elements in the x and y directions. Symmetry boundaries are on the $x = 0$ and $y = 0$ planes and an outflow boundary is on the other outer surfaces. Data for this test is taken along a diagonal line characterized by the vector $\{1, 1\}$.

Test 7, Figure 3.44(g), uses a three-dimensional mesh to compute the 1D spherical Noh test. It consists of a Cartesian block of elements, uniformly spaced with n elements in the x , y , and z directions. Symmetry boundaries are on the $x = 0$, $y = 0$, and $z = 0$ planes and an outflow boundary is on the other outer surfaces. Data for this test is taken along a diagonal line characterized by the vector $\{1, 1, 1\}$.

Test Theory

The Noh test has an analytical solution. Noh [85] provided a general analytical closed-form solution to the problem. The theoretical solution was computed using methods discussed in [83].

Test Results

The computations of the seven tests were performed on a fixed Eulerian mesh using both the first-order method (predictor off) and the second-order scheme (predictor on). The results are plotted in Figures 3.45–3.51 for the first-order method, while second-order results are plotted in Figures 3.52–3.58. The numerical results for the various grid resolutions are given by the colored symbols, while the exact solution is represented by the solid black line. The tests were run using four mesh resolutions, $n = 30, 60, 120, 240$, where n is the number of grid cells in each coordinate direction. Thus for Test 7, when $n = 240$ there are a total of $240^3 = 13,824,000$ computational cells. Generally, it appears that the simulations are converging to the theoretical value as the mesh resolution improves.

The results for the first-order method are fairly consistent for each geometry. Accurately predicting the density behind the shock is a challenge for hydrocodes. The computed density in the current calculations is generally under-predicted, but improves with resolution. Although the theoretical state behind the shock should be constant, the current results show lots of gradients. It is not understood why this is, but that region appears under-resolved so additional mesh resolution could help. The shock speed appears to be over-predicted using the first-order method. Turning the predictor on appears to improve the predicted shock location.

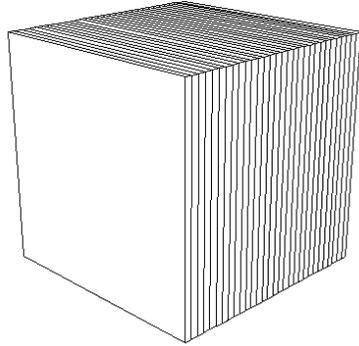
Similar to the Sedov test results in §3.1.4, the results for the Noh test on multidimensional meshes are expected to have some angular variation in the solution. The extent of the angular variation in the data can be seen in the scatter plots shown in Figures 3.59–3.64. Unlike the earlier plots, where data was taken along a specified line segment, the scatter plots gather data at every computational cell and plot it versus the radial position. Although there appears

to be some scatter in the data, there is not as much angular variation as seen in the Sedov tests. It also appears that angular variation is reduced when the predictor is turned on.

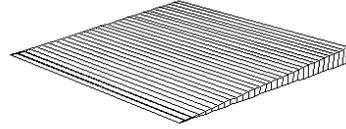
Test Conclusions

The following conclusions can be drawn from this calculation:

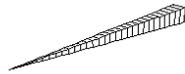
- The calculations show that the current code can be used to capture the implosion associated with the Noh test.
- Although the shock appears to be propagating properly in the current results, the resulting state behind the shock is full of gradients. Therefore, it is difficult to ascertain the accuracy of the method in those regions. It is assumed that additional grid refinement would improve these calculations.
- Scatter plots reveal that there is some angular variation in the solution for multidimensional meshes, but it is small compared to the Sedov test results.



(a) Test 1



(b) Test 2

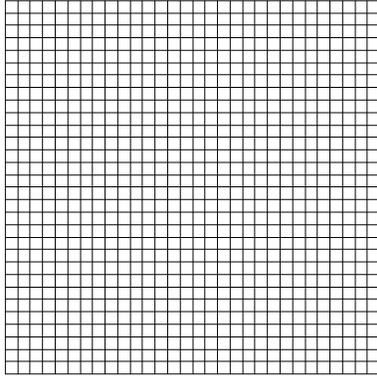


(c) Test 3

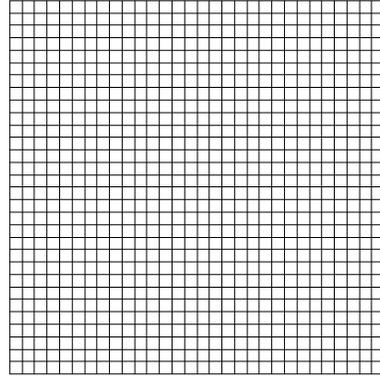


(d) Test 4

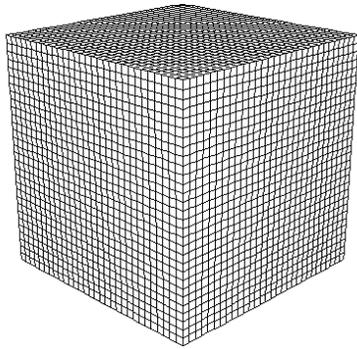
Figure 3.44: Initial mesh geometries for the Noh tests.



(e) Test 5

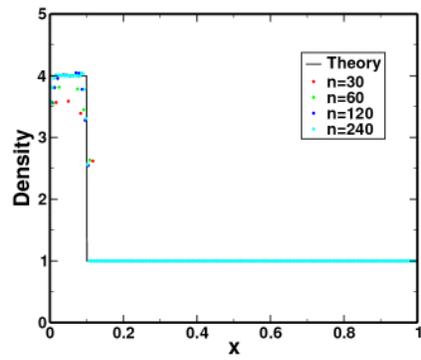


(f) Test 6

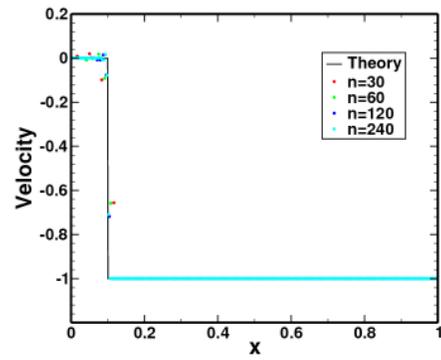


(g) Test 7

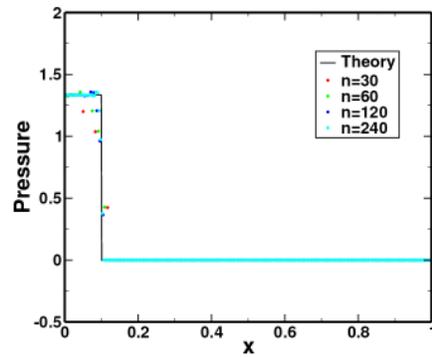
Figure 3.44: Initial mesh geometries for the Noh tests. (cont)



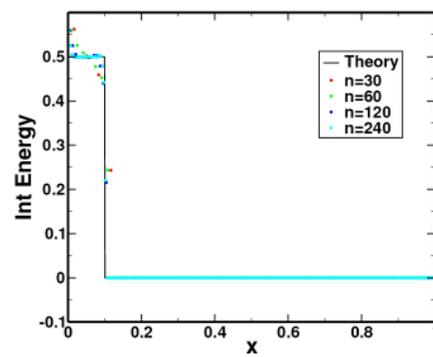
(a) density



(b) velocity



(c) pressure



(d) energy

Figure 3.45: Single-phase numerical results for the Noh Test #1 (1D Cartesian geometry). The tests used an *Eulerian grid* with the *predictor off*.

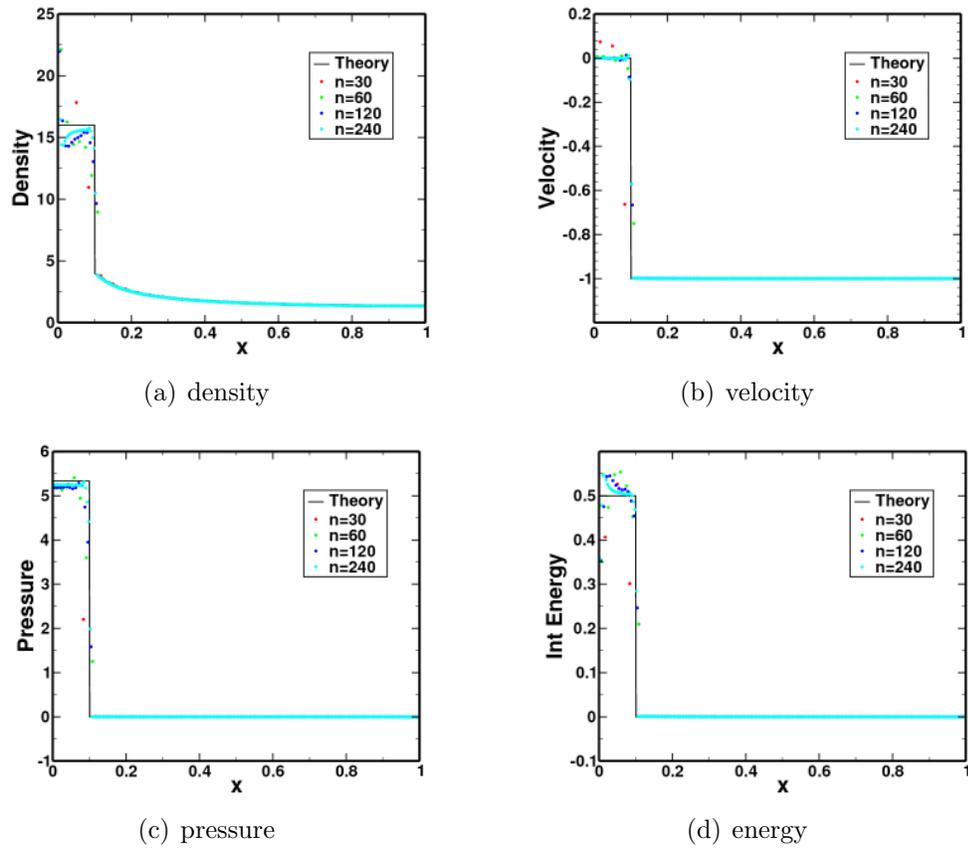


Figure 3.46: Single-phase numerical results for the Noh Test #2 (1D Cylindrical geometry). The tests used an *Eulerian grid* with the *predictor off*.

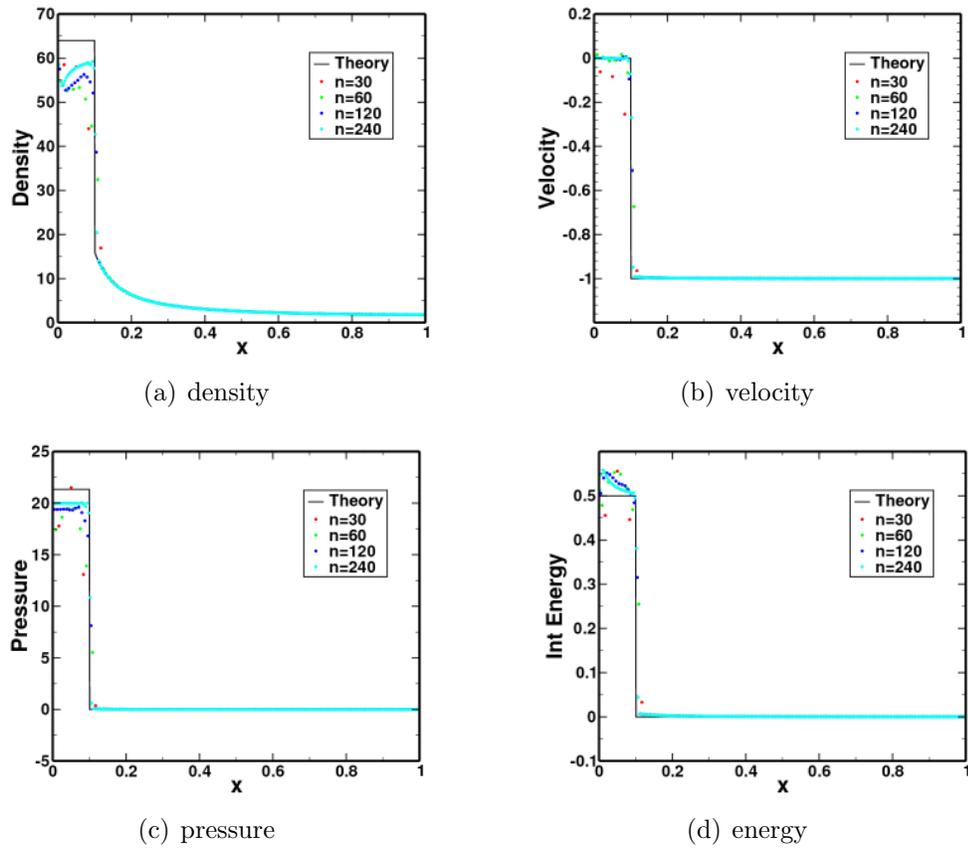


Figure 3.47: Single-phase numerical results for the Noh Test #3 (1D Spherical geometry). The tests used an *Eulerian grid* with the *predictor off*.

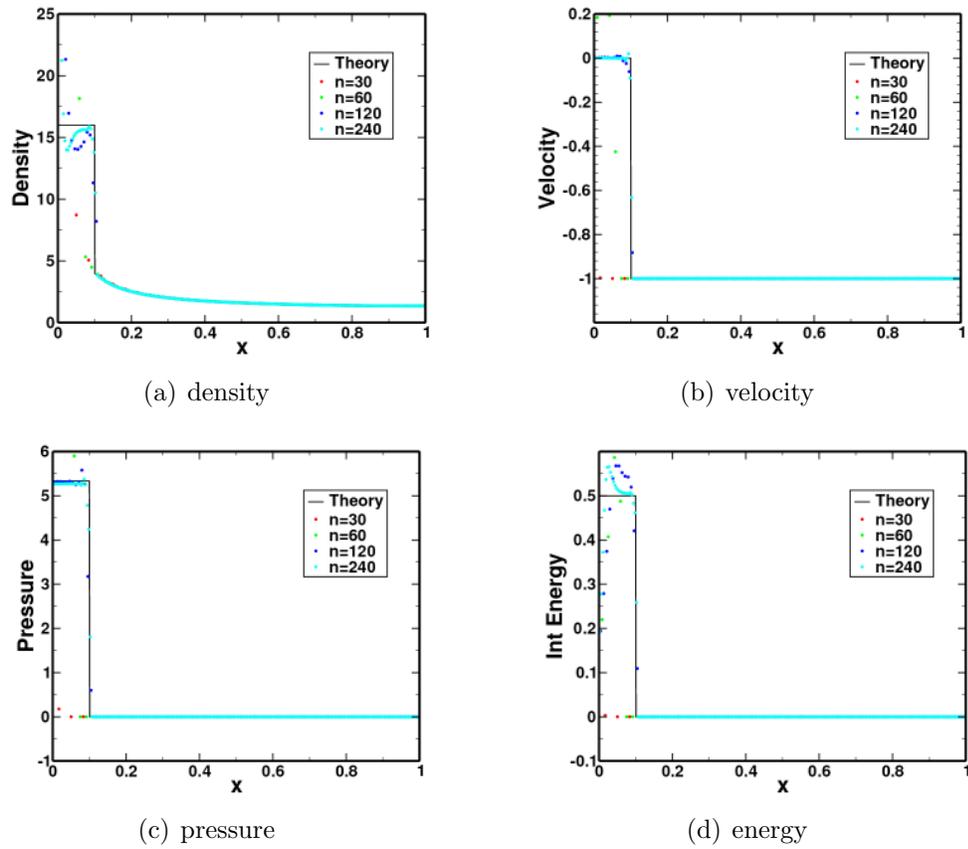


Figure 3.48: Single-phase numerical results for the Noh Test #4 (1D-axisymmetric Cylindrical geometry). The tests used an *Eulerian grid* with the *predictor off*.

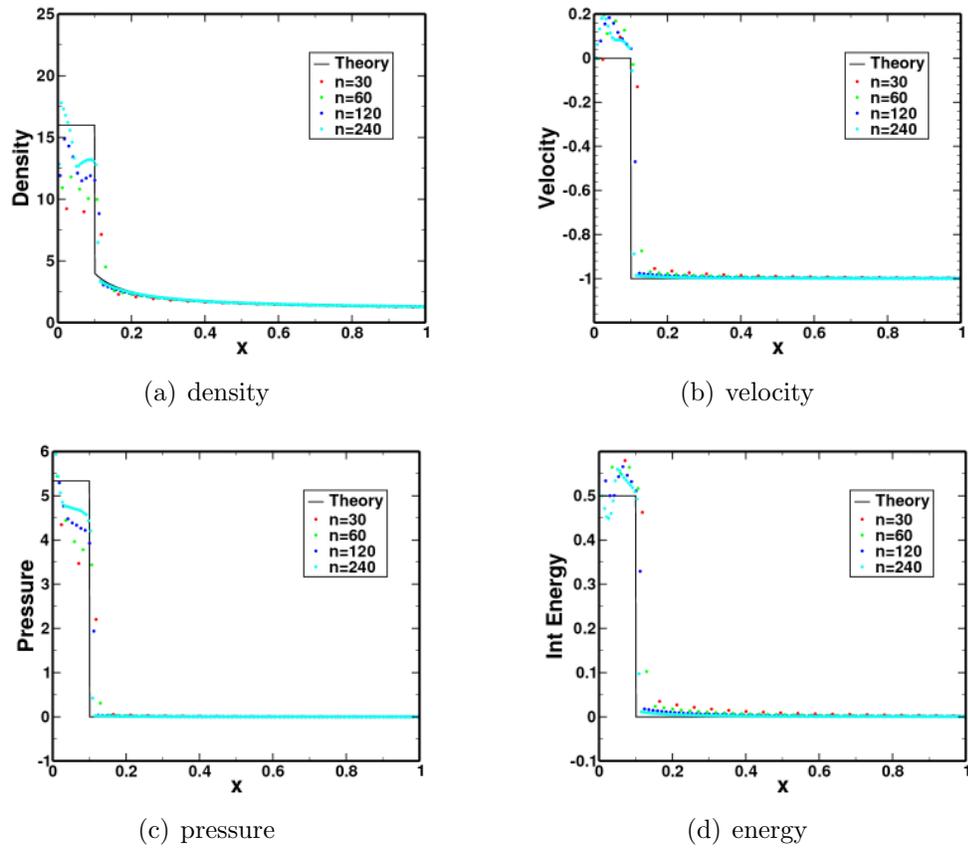


Figure 3.49: Single-phase numerical results for the Noh Test #5 (2D Cylindrical geometry). The tests used an *Eulerian grid* with the *predictor off*.

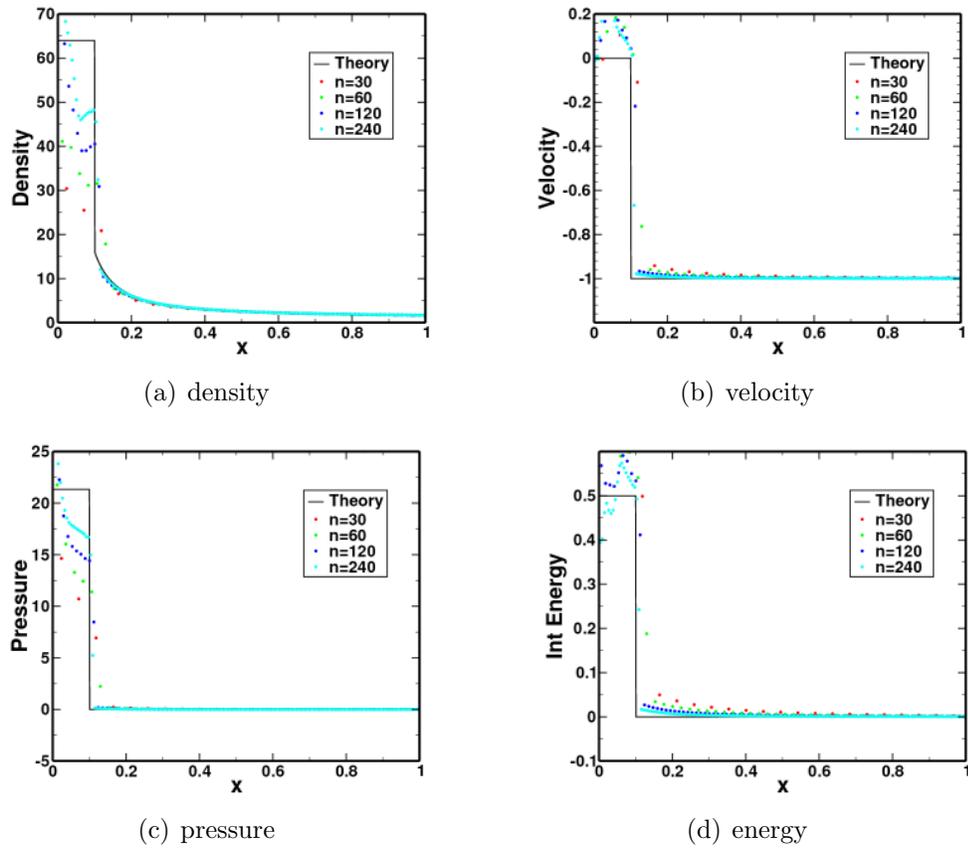


Figure 3.50: Single-phase numerical results for the Noh Test #6 (2D-axisymmetric Spherical geometry). The tests used an *Eulerian grid* with the *predictor off*.

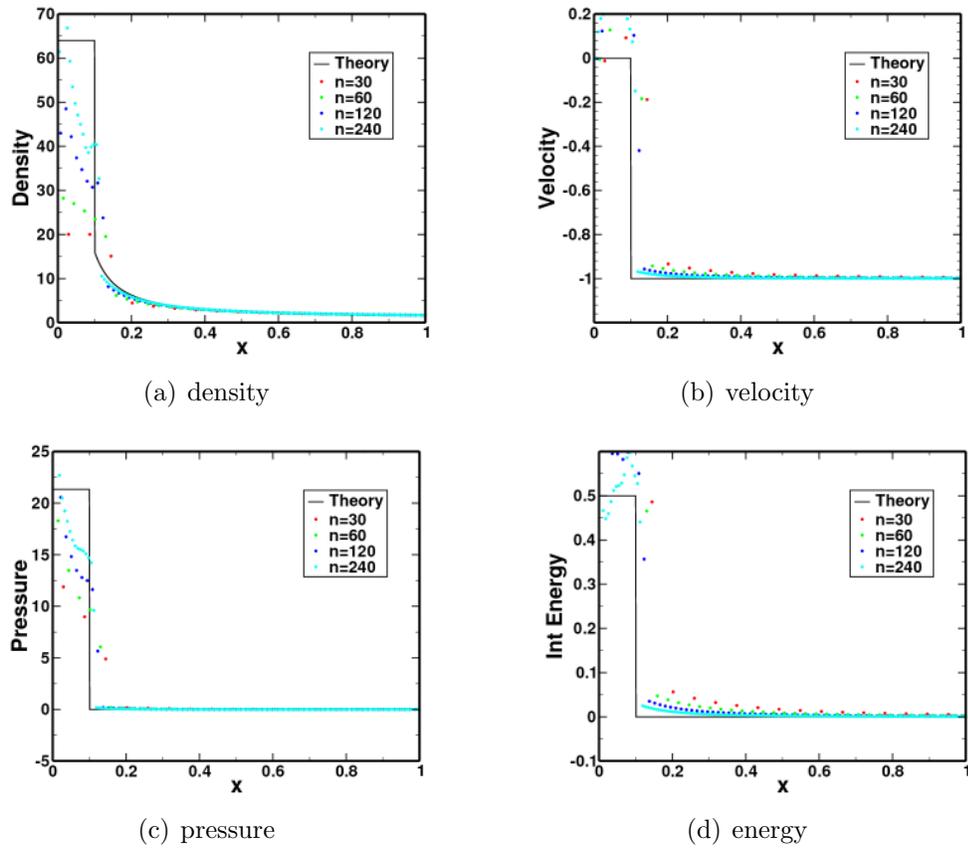
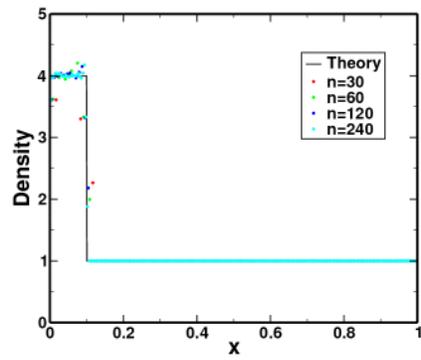
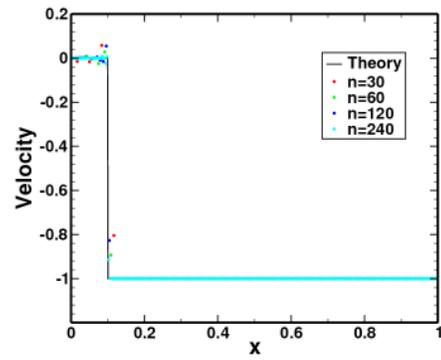


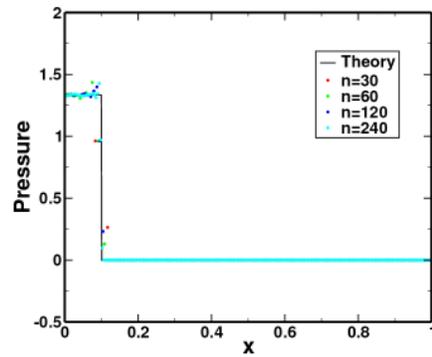
Figure 3.51: Single-phase numerical results for the Noh Test #7 (3D Spherical geometry). The tests used an *Eulerian grid* with the *predictor off*.



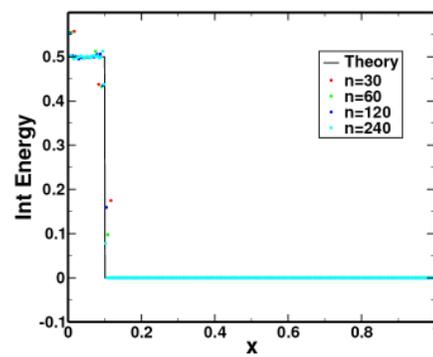
(a) density



(b) velocity



(c) pressure



(d) energy

Figure 3.52: Single-phase numerical results for the Noh Test #1 (1D Cartesian geometry). The tests used an *Eulerian grid* with the *predictor on*.

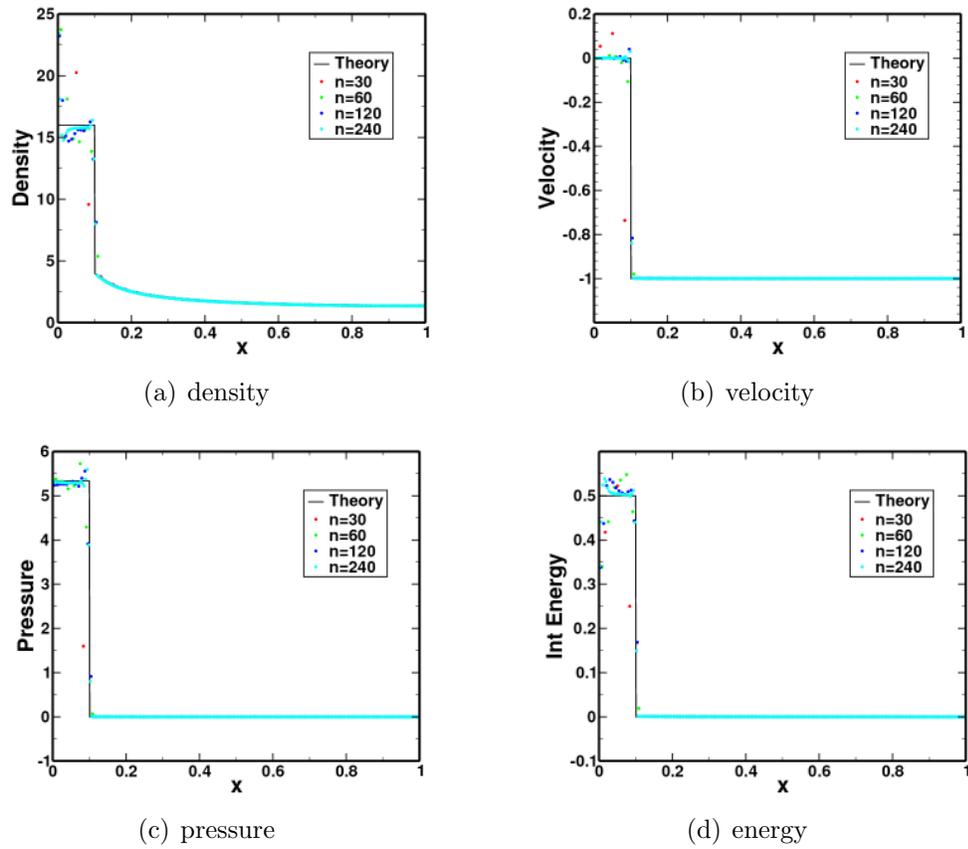


Figure 3.53: Single-phase numerical results for the Noh Test #2 (1D Cylindrical geometry). The tests used an *Eulerian grid* with the *predictor on*.

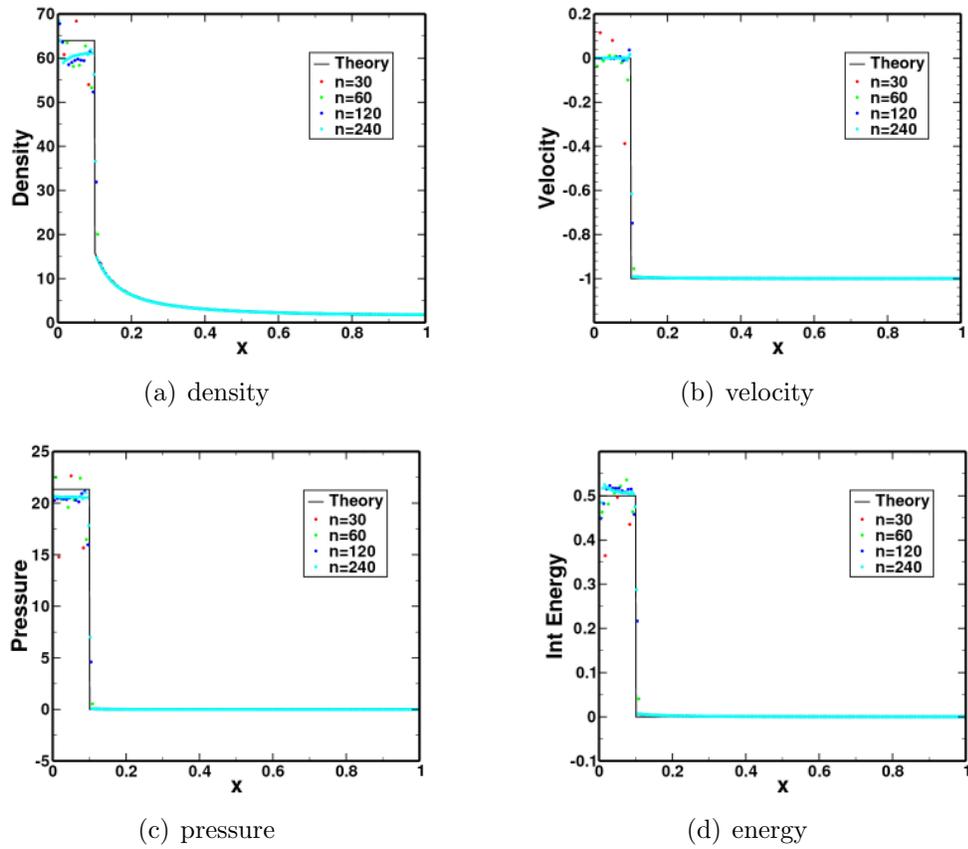


Figure 3.54: Single-phase numerical results for the Noh Test #3 (1D Spherical geometry). The tests used an *Eulerian grid* with the *predictor on*.

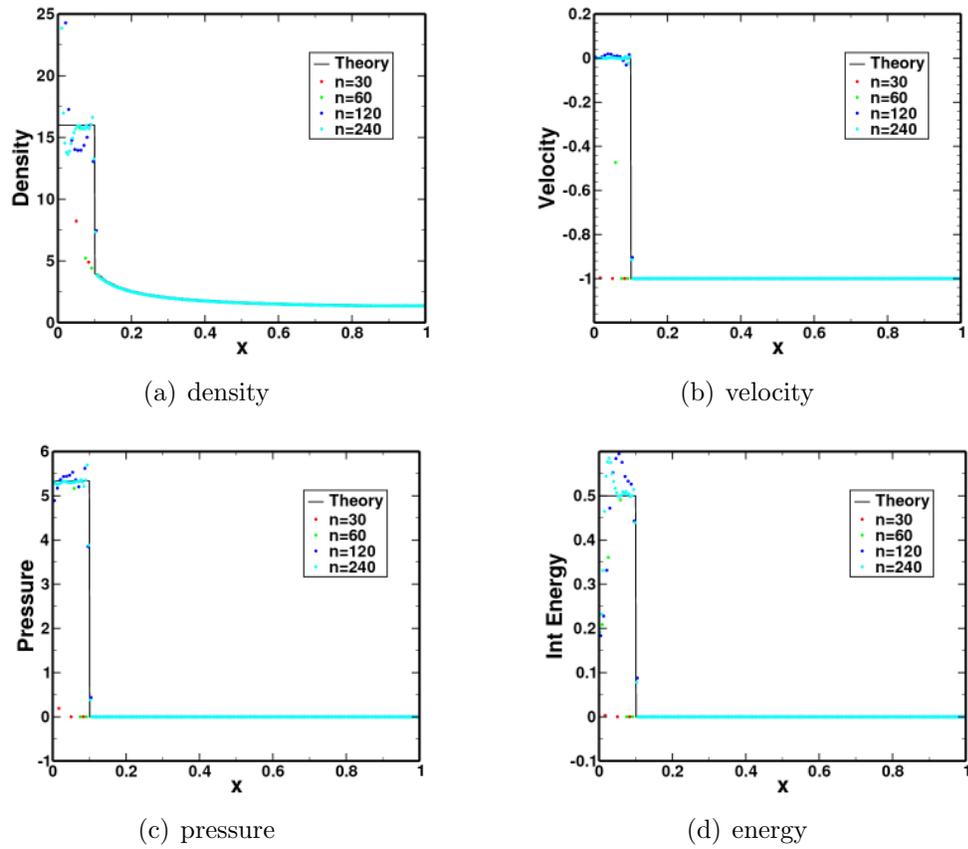


Figure 3.55: Single-phase numerical results for the Noh Test #4 (1D-axisymmetric Cylindrical geometry). The tests used an *Eulerian grid* with the *predictor on*.

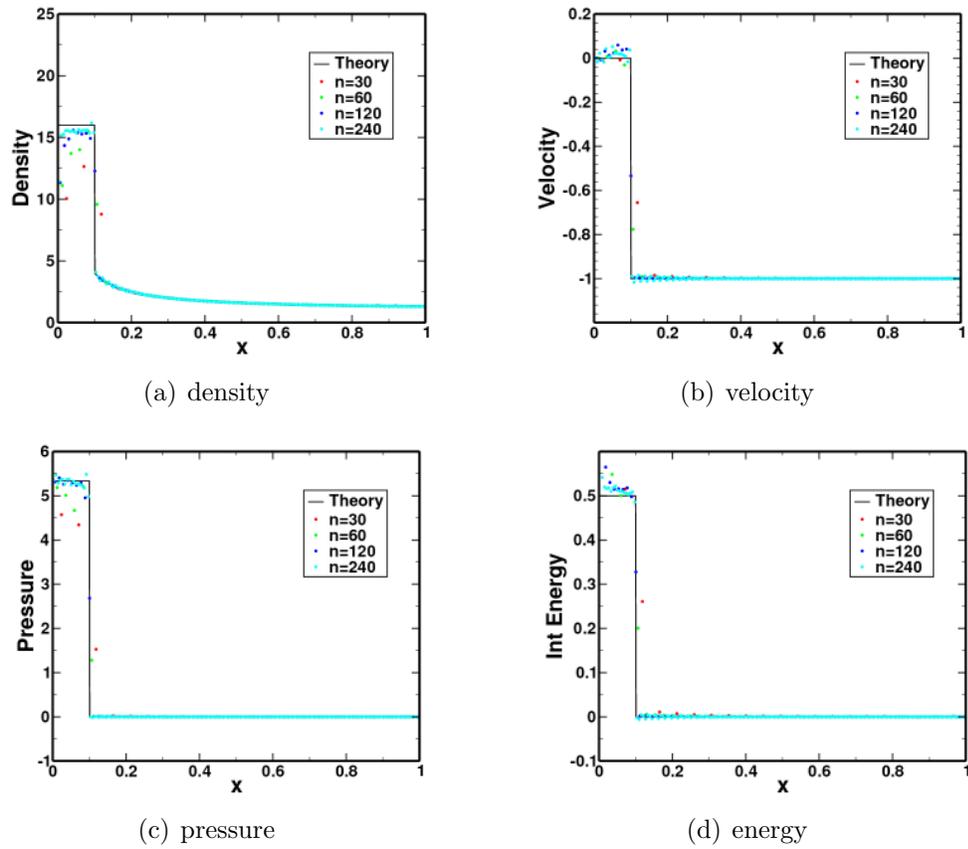


Figure 3.56: Single-phase numerical results for the Noh Test #5 (2D Cylindrical geometry). The tests used an *Eulerian grid* with the *predictor on*.

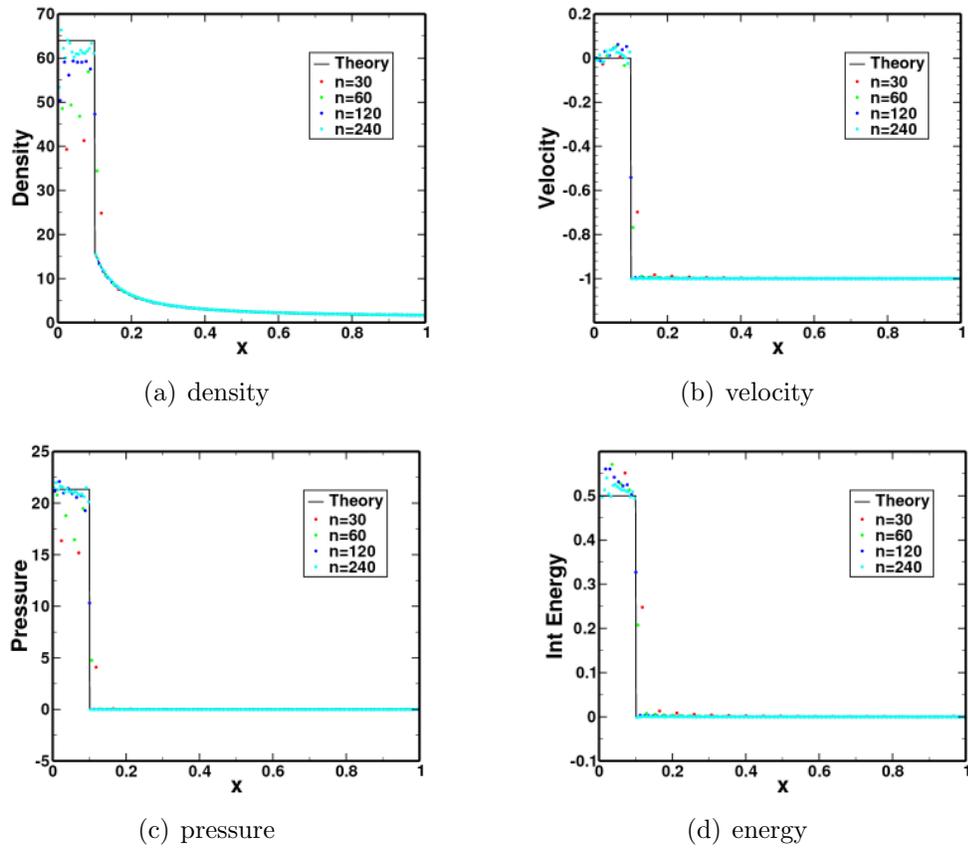


Figure 3.57: Single-phase numerical results for the Noh Test #6 (2D-axisymmetric Spherical geometry). The tests used an *Eulerian grid* with the *predictor on*.

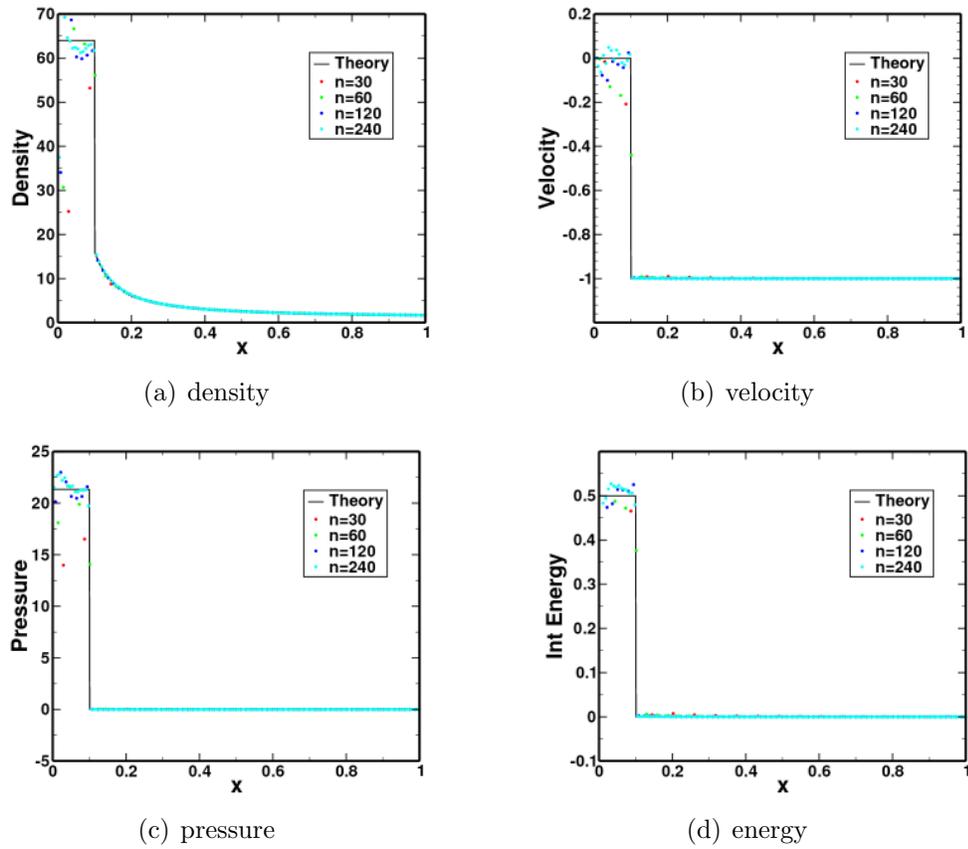


Figure 3.58: Single-phase numerical results for the Noh Test #7 (3D Spherical geometry). The tests used an *Eulerian grid* with the *predictor on*.

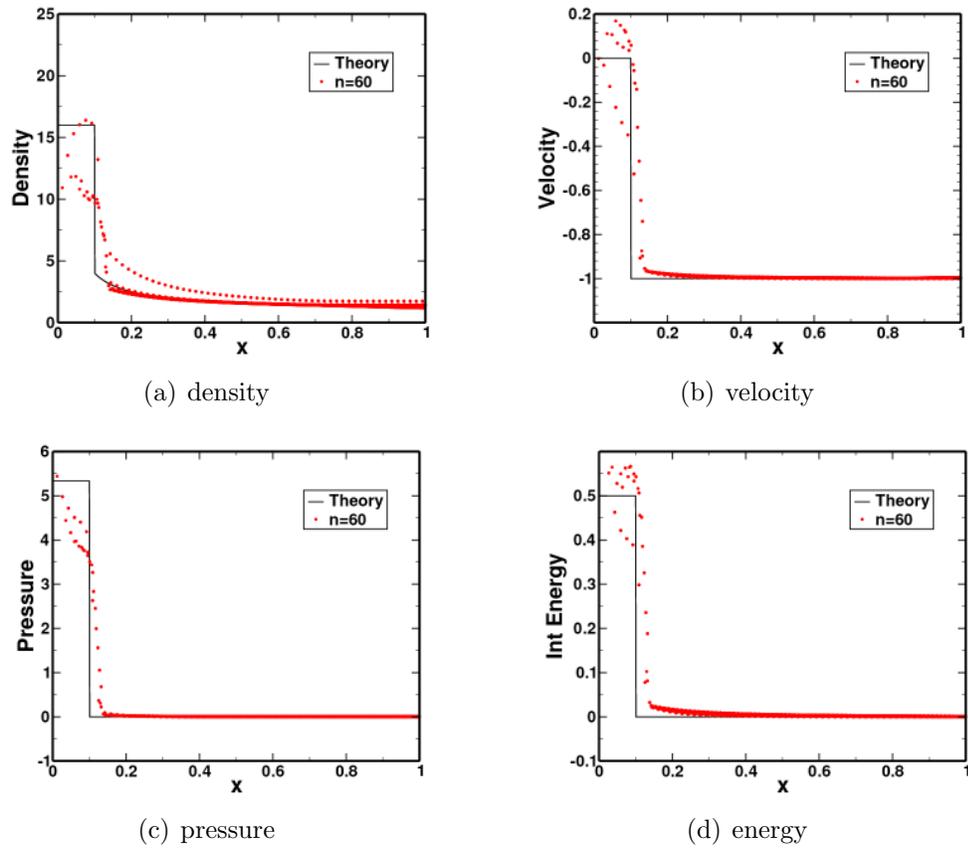


Figure 3.59: Single-phase numerical scatter-plot results for the Noh Test #5 (2D Cylindrical geometry). The tests used an *Eulerian grid* with the *predictor off*.

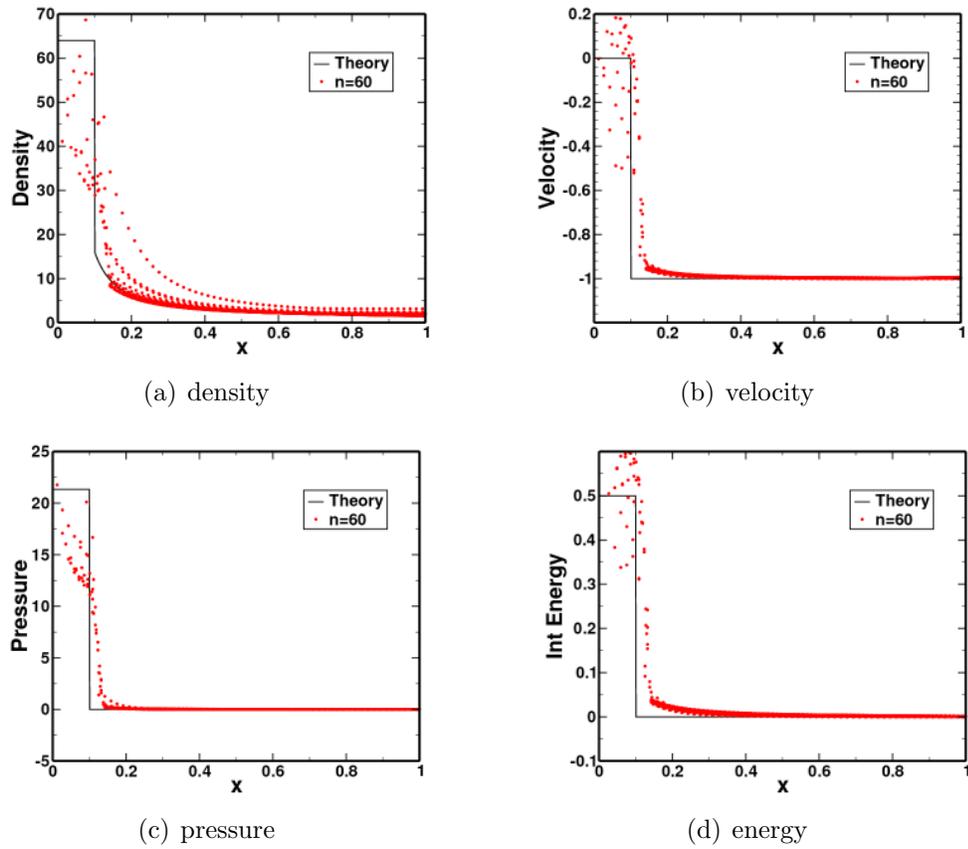


Figure 3.60: Single-phase numerical scatter-plot results for the Noh Test #6 (2D-axisymmetric Spherical geometry). The tests used an *Eulerian grid* with the *predictor off*.

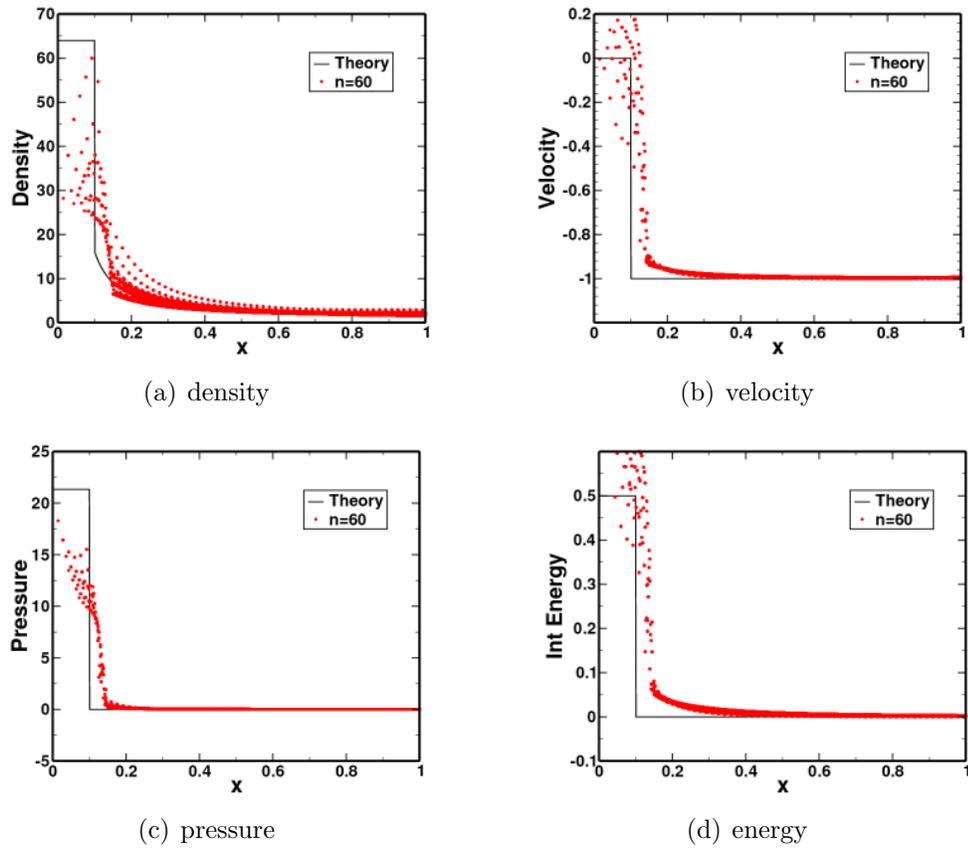


Figure 3.61: Single-phase numerical scatter-plot results for the Noh Test #7 (3D Spherical geometry). The tests used an *Eulerian grid* with the *predictor off*.

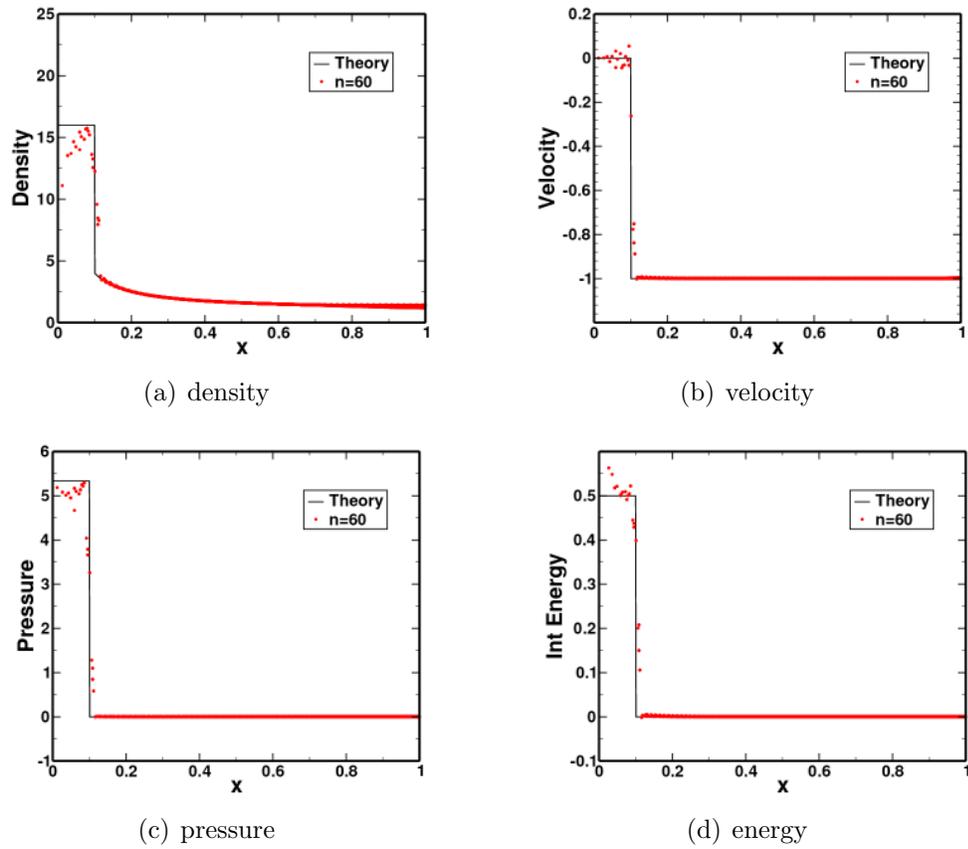


Figure 3.62: Single-phase numerical scatter-plot results for the Noh Test #5 (2D Cylindrical geometry). The tests used an *Eulerian grid* with the *predictor on*.

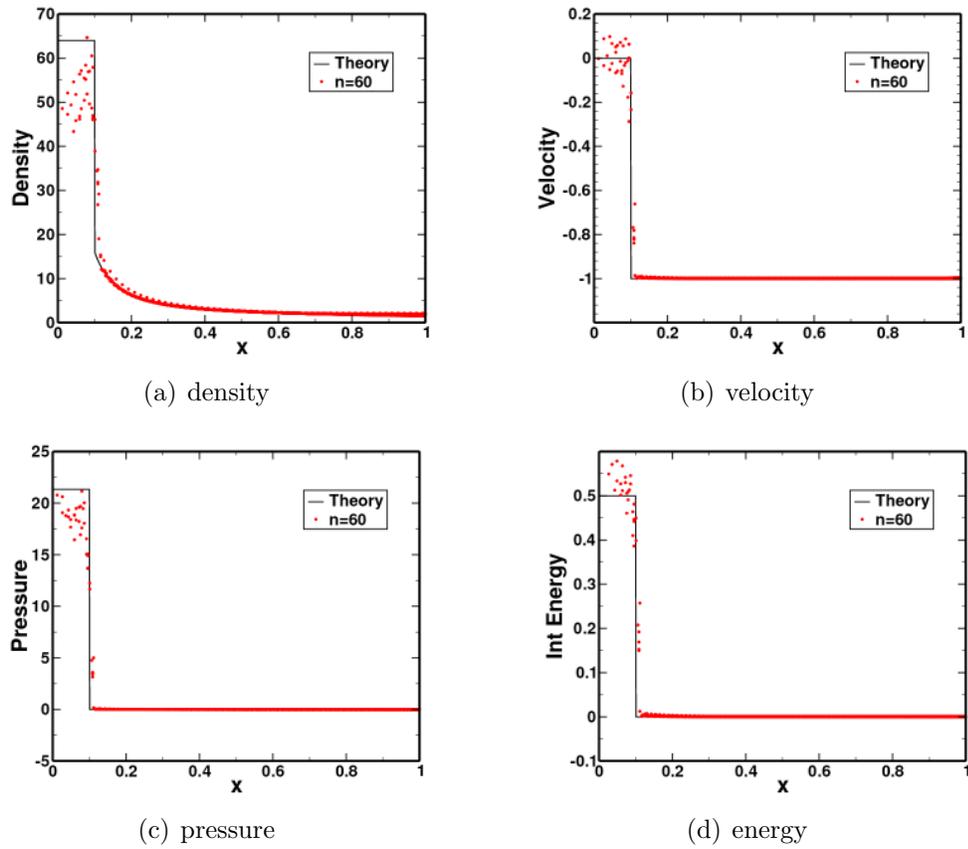


Figure 3.63: Single-phase numerical scatter-plot results for the Noh Test #6 (2D-axisymmetric Spherical geometry). The tests used an *Eulerian grid* with the *predictor on*.

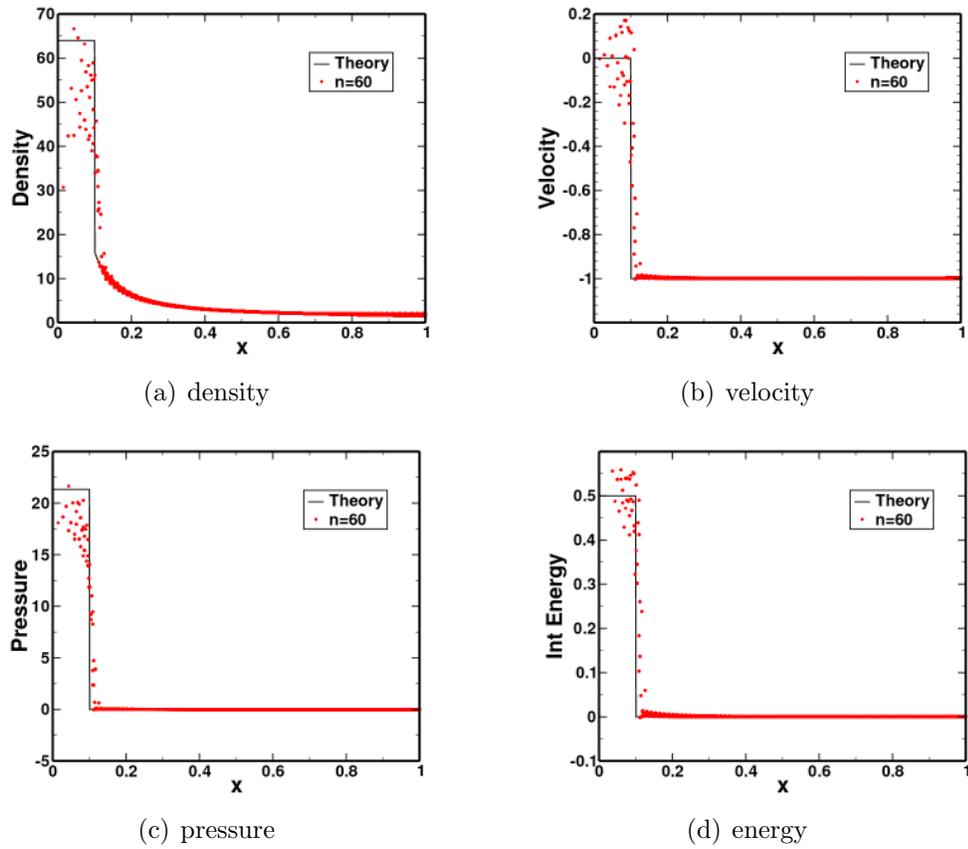


Figure 3.64: Single-phase numerical scatter-plot results for the Noh Test #7 (3D Spherical geometry). The tests used an *Eulerian grid* with the *predictor on*.

3.2 Multiple Phases

The following section exercises the current model on a number of tests with multiple phases. The tests evaluate the model for use as a diffuse-interface model as well as modeling sub-grid particulate flow. The tests are designed to exercise the non-conservative terms in the governing equations as well as the relaxation terms. The tests are set up to isolate specific multiphase features and evaluate them individually.

The tests begin with a convergence study on the numerical method. This is similar to the single-phase convergence test, but uses multiple phases. This is followed by a couple shock-tube problems involving multiple materials. Each material is treated as a separate phase, resulting in a diffuse interface between the materials. Next are some tests of the relaxation models, which isolate and evaluate the model's heat-transfer and drag formulations. The drag model is next evaluated for the propagation of a particle cloud through a gas. Finally, the model is compared to experimental data of a shock wave passing through a gas-particle mixture.

3.2.1 Multiphase Smooth Test Problem

Test Description

The following test is used to verify the order of accuracy of the method for a two-phase mixture. It is a numerical exercise and not designed to model any physical reality. One of the implicit assumptions when discussing the order of a numerical method is that the solution is *smooth*. Therefore, this test uses smooth initial conditions. Chosen appropriately, smooth spatial variations at the beginning of the run will result in transient behavior that remains smooth throughout the simulation. Simulations are performed on a series of grids in order to compute the convergence rate of the method. An example of a similar test in the literature is found in Schwendeman et al. [48].

This test is similar to the problem discussed in §3.1.1. Whereas the first test was limited to a single-phase material, the current test demonstrates the order of accuracy for a simulation of two phases.

Test Setup

This test is designed to exercise the current multiphase algorithm for a two-phase material, designated as phase a and phase b . Both phase materials use an ideal-gas equation of state with $\gamma_a = \gamma_b = 1.4$. The material state is initialized using the functions

$$\varepsilon_a = 2.5$$

$$\varepsilon_b = 2.5$$

$$p_a = 1.0$$

$$p_b = 1.0$$

$$x_a = 0.5 - 0.1 \tanh(20x - 8)$$

$$x_b = 0.5 + 0.1 \tanh(20x - 8)$$

$$u_a = 0.1 - 0.2 \tanh(20x - 12)$$

$$u_b = 0.2 + 0.3 \tanh(20x - 10)$$

which are smooth on the domain of interest. The functional form of the initial conditions is somewhat arbitrary, but they were chosen in order to obtain smoothly varying transient behavior. The above conditions prescribe a varying volume fraction, thus the phases are coupled and the non-conservative terms will contribute to the dynamic response. No relaxation terms were used in these calculations, i.e. phase interaction due to pressure relaxation, drag, and thermal conduction were neglected.

The problem is designed to be one dimensional. It is modeled using a three-dimensional mesh with a single zone in the y and z directions and symmetry boundary conditions on the sides, resulting in a pseudo-1D flow in the x direction, with $x \in [0, 1]$. The number of zones in the x direction are varied to look at mesh convergence. Outflow boundary conditions are given at the boundaries $x = 0$ and $x = 1$. An Eulerian mesh, with no mesh motion, is used for the simulations.

The solutions are compared at the simulation end time $t = 0.05$. A constant time-step was used for these tests. The size of the time-step was chosen to be $\Delta t = \frac{0.05}{N}$, where N is the number of grid cells. This was found to satisfy the stability conditions at all times.

Test Theory

No analytic solution is available for this test. Smooth analytic solutions are difficult to obtain in general. However, the exact solution can be approximated with sufficient accuracy using the current numerical method with a very large number of grid cells. For the current study 12,800 grid cells were used to obtain the baseline solution. Baseline solutions were obtained using both the first-order and the second-order algorithms. The first-order solution is used as the baseline for the first-order results on the coarser meshes, and the second-order solution is used as the baseline for the second-order results.

Test Results

The final state of each phase was calculated using both the first-order and second-order schemes. The first-order (predictor off) results are plotted in Figure 3.65,

and the second-order (predictor on) results are plotted in Figure 3.66. Calculations using four mesh-refinement levels were performed using 100, 200, 400, and 800 grid cells. The numerical results for the various grid resolutions are given by the colored symbols, while the *high-resolution* solution is designated as *Theory* and represented by the solid black line.

All of the calculations resulted in a smooth final flow-field with no shocks or contact discontinuities developing. The results also reveal spatial variation in all of the plotted variables. It appears that the calculations are converging towards an answer as the mesh is refined. The first-order solutions exhibit more variation between the finest and coarsest grids than the second-order method.

Test Error Quantification and Convergence

The discretization error for the solution can be quantified using the fine-grid solution as a baseline. The error (E_N) is determined for a mesh-refinement level N using the methods of §E. The calculated error for the current solutions has been computed for the variables density, velocity, pressure, internal energy, and volume fraction of both phases.

The errors are tabulated along with the associated convergence rates (r_N) in Table 3.7 for the first-order method. The results indicate that the first-order scheme is indeed first-order since the convergence rates are very close to one.

The errors for the second-order method are shown in Table 3.8. The convergence rate for a second-order scheme should have a value of two. However, the current results point toward a convergence rate closer to one, thus indicating first-order convergence rather than second-order. It is noted that the magnitude of the error is lower for the second-order method than for the first-order.

In the single-phase version of this test (§3.1.1), second-order spatial convergence was obtained by making a couple of modifications to the tests. These same modifications were also tried on the current test. In the new tests, the time-step was given a constant value of $\Delta t = \frac{0.05}{12800}$ for all grid levels. Also, the new tests were run with the predictor on, but the slope limiters were disabled. The

observed error and convergence rates for these new tests are tabulated in Table 3.9. As seen, the test modifications have improved the convergence rates slightly, but they are still far below second-order.

It is not currently understood why the second-order scheme fails to demonstrate second-order convergence in this test, even after disabling the slope limiters. Since the second-order spatial convergence was observed in the single-phase version of this test, it is assumed that the cause is related to something specific within the multiphase coupling terms. A possibility is that the use of the simplified predictor (2.86) is causing the rate of convergence to drop. Perhaps using the full predictor (2.85), including contributions from the multiphase nozzling terms, would regain the desired order of accuracy. Resolving this issue will be an important component of future research.

Test Conclusions

A test problem utilizing smooth initial conditions has been used to assess the order of accuracy of the current method for a two-phase flow. The following conclusions can be drawn from this calculation:

- The current first-order scheme demonstrates an observed convergence rate of first-order.
- The second-order scheme does not demonstrate second-order convergence, but rather exhibits convergence rates closer to first-order. Disabling the slope limiters in the scheme appear to improve the convergence rates, but they are still far below second order. The reasons for this are not currently known, but some possible explanations have been proposed.
- The current test relies on one-dimensional geometries. It would be straightforward, although more computationally expensive, to extend the current test to three dimensions.
- A more rigorous (and more complicated) technique for order-of-accuracy verification is the method of manufactured solutions as discussed by

Roache [78] and Salari et al. [79]. The idea here is to manufacture an exact solution and then modify the original governing equations by adding source terms in such a way that the manufactured function becomes the exact solution. This could be a very useful way to track down why the current algorithm is not achieving second-order accuracy.

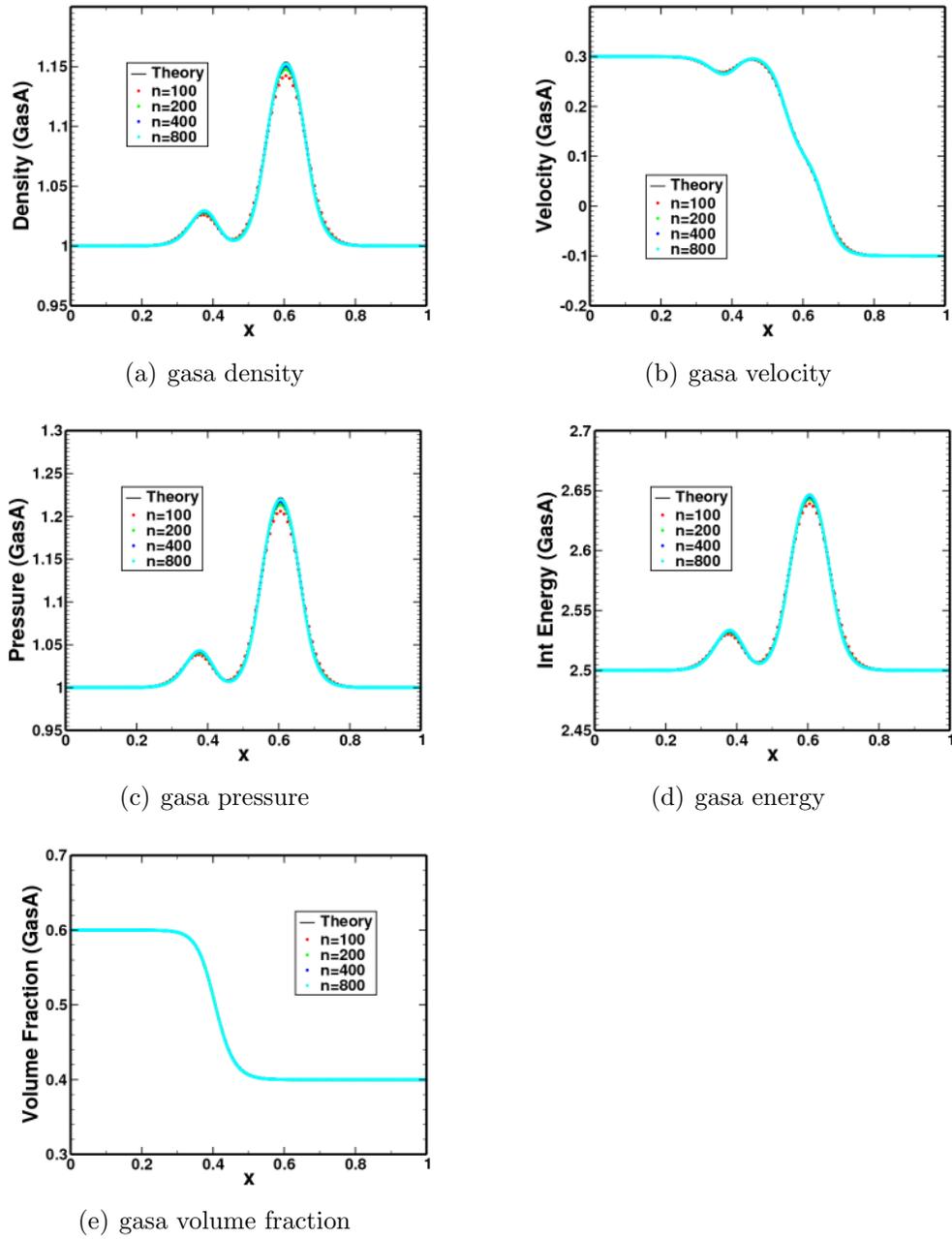
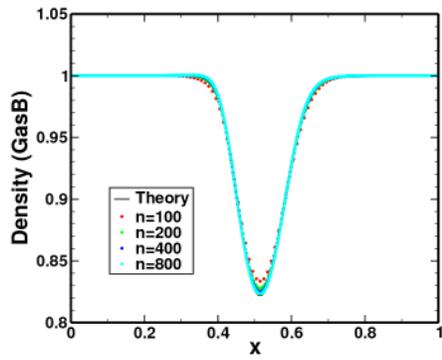
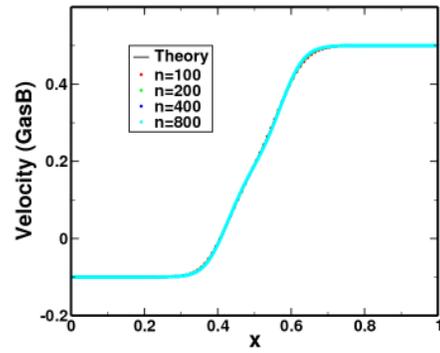


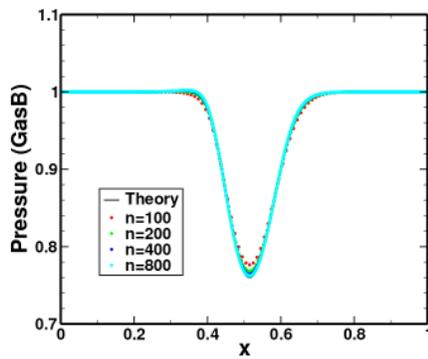
Figure 3.65: Numerical results for the multiphase smooth test. The tests used an *Eulerian grid* with the *predictor off*.



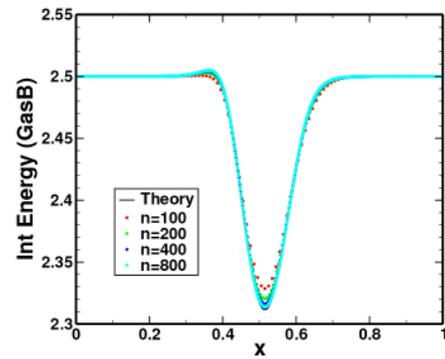
(f) gasb density



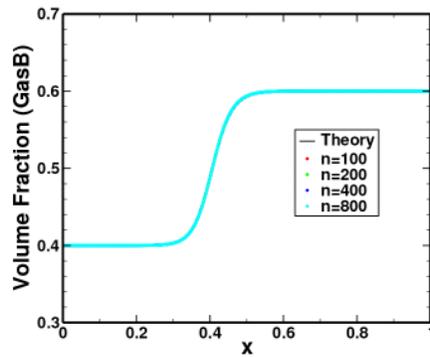
(g) gasb velocity



(h) gasb pressure



(i) gasb energy



(j) gasb volume fraction

Figure 3.65: Numerical results for the multiphase smooth test. The tests used an *Eulerian grid* with the *predictor off*. (cont)

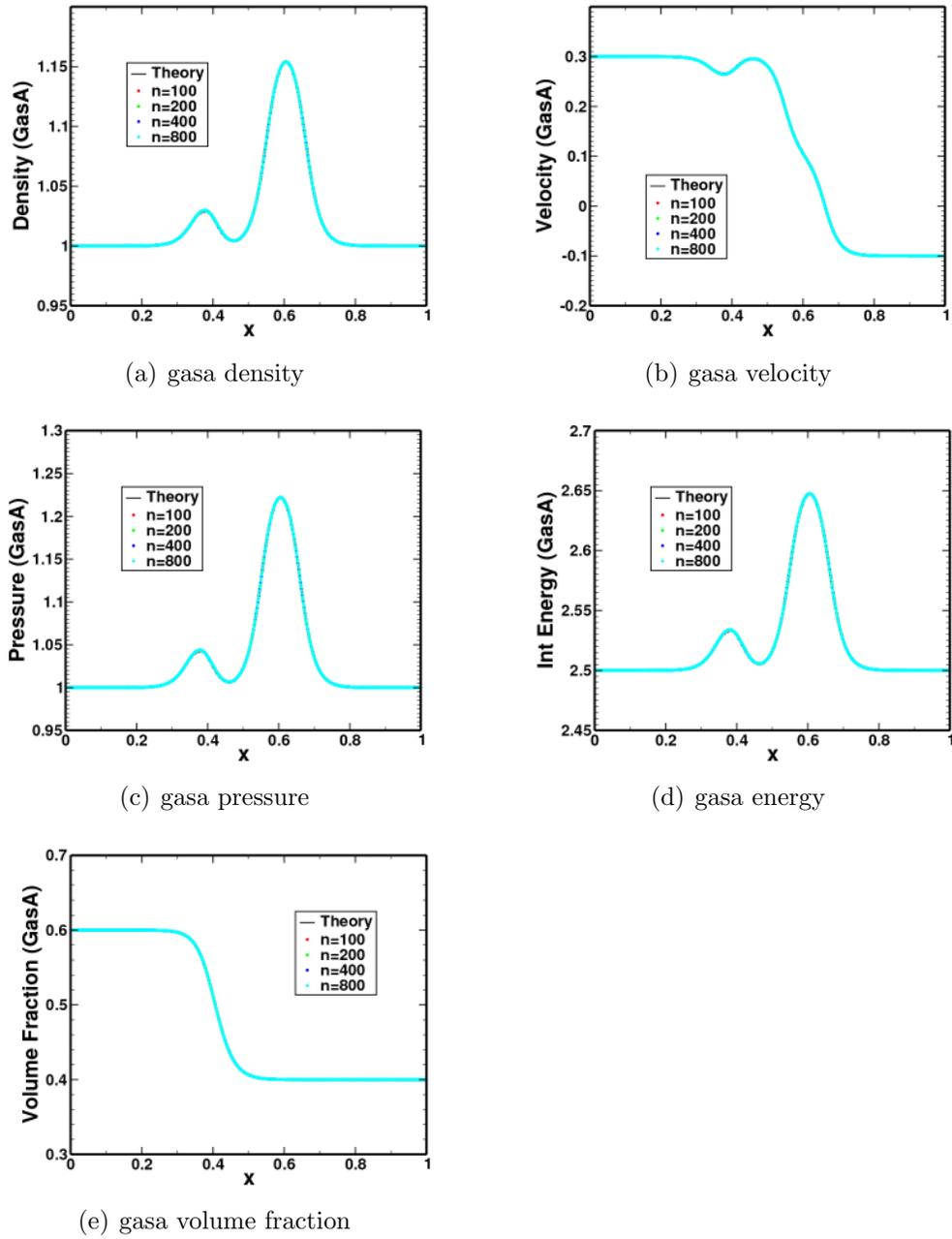
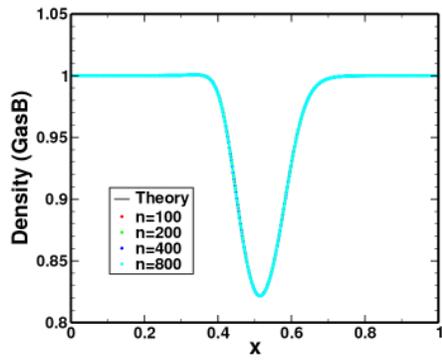
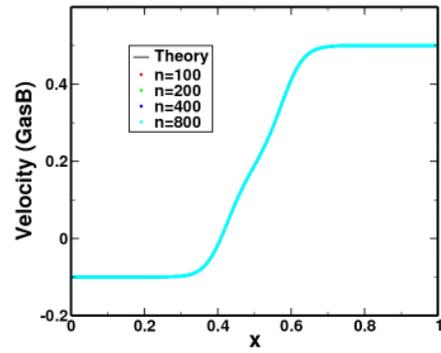


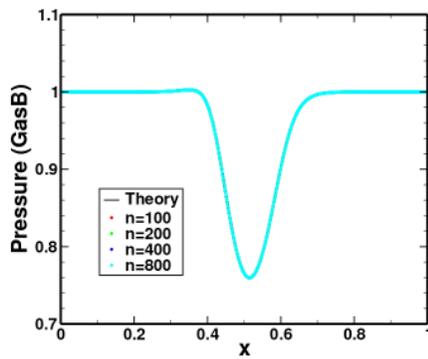
Figure 3.66: Numerical results for the multiphase smooth test. The tests used an *Eulerian grid* with the *predictor on*.



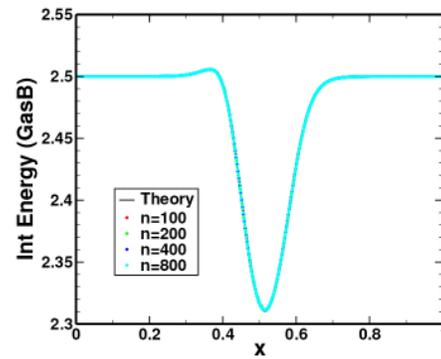
(f) gasb density



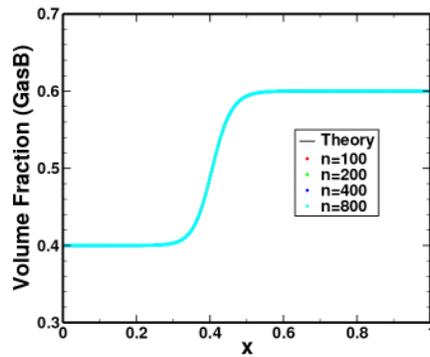
(g) gasb velocity



(h) gasb pressure



(i) gasb energy



(j) gasb volume fraction

Figure 3.66: Numerical results for the multiphase smooth test. The tests used an *Eulerian grid* with the *predictor on*. (cont)

N	E_N	r_N
Gas A Density		
100	3.82e-03	
200	2.02e-03	0.92
400	1.03e-03	0.97
800	5.10e-04	1
Gas A Velocity		
100	3.18e-03	
200	1.72e-03	0.89
400	8.86e-04	0.96
800	4.39e-04	1
Gas A Pressure		
100	5.37e-03	
200	2.84e-03	0.92
400	1.45e-03	0.97
800	7.16e-04	1
Gas A Energy		
100	3.29e-03	
200	1.74e-03	0.92
400	8.86e-04	0.97
800	4.37e-04	1
Gas A Mass Fraction		
100	1.00e-03	
200	5.22e-04	0.94
400	2.64e-04	0.98
800	1.30e-04	1
Gas A Volume Fraction		
100	3.46e-04	
200	1.73e-04	1
400	8.58e-05	1
800	4.16e-05	1

Table 3.7: Numerical error and convergence rates for the multiphase smooth test. The tests used an *Eulerian grid* with the *predictor off*.

N	E_N	r_N
Gas B Density		
100	4.21e-03	
200	2.19e-03	0.94
400	1.11e-03	0.98
800	5.43e-04	1
Gas B Velocity		
100	3.41e-03	
200	1.81e-03	0.91
400	9.26e-04	0.97
800	4.57e-04	1
Gas B Pressure		
100	5.94e-03	
200	3.09e-03	0.94
400	1.56e-03	0.98
800	7.67e-04	1
Gas B Energy		
100	5.57e-03	
200	2.90e-03	0.94
400	1.47e-03	0.98
800	7.21e-04	1
Gas B Mass Fraction		
100	1.00e-03	
200	5.22e-04	0.94
400	2.64e-04	0.98
800	1.30e-04	1
Gas B Volume Fraction		
100	3.46e-04	
200	1.73e-04	1
400	8.58e-05	1
800	4.16e-05	1

Table 3.7: Numerical error and convergence rates for the multiphase smooth test. The tests used an *Eulerian grid* with the *predictor off*. (cont)

N	E_N	r_N
Gas A Density		
100	3.69e-04	
200	1.60e-04	1.2
400	8.15e-05	0.98
800	4.12e-05	0.99
Gas A Velocity		
100	4.56e-04	
200	1.86e-04	1.3
400	8.88e-05	1.1
800	4.32e-05	1
Gas A Pressure		
100	5.37e-04	
200	2.35e-04	1.2
400	1.19e-04	0.99
800	6.01e-05	0.98
Gas A Energy		
100	3.80e-04	
200	1.76e-04	1.1
400	8.83e-05	0.99
800	4.39e-05	1
Gas A Mass Fraction		
100	1.12e-04	
200	5.45e-05	1
400	2.74e-05	0.99
800	1.37e-05	1
Gas A Volume Fraction		
100	1.08e-04	
200	5.61e-05	0.95
400	2.88e-05	0.96
800	1.43e-05	1

Table 3.8: Numerical error and convergence rates for the multiphase smooth test. The tests used an *Eulerian grid* with the *predictor on*.

N	E_N	r_N
Gas B Density		
100	2.72e-04	
200	1.63e-04	0.74
400	9.18e-05	0.83
800	4.74e-05	0.95
Gas B Velocity		
100	4.82e-04	
200	2.21e-04	1.1
400	1.06e-04	1.1
800	5.18e-05	1
Gas B Pressure		
100	3.70e-04	
200	2.25e-04	0.71
400	1.27e-04	0.82
800	6.60e-05	0.95
Gas B Energy		
100	3.18e-04	
200	1.84e-04	0.79
400	1.03e-04	0.84
800	5.33e-05	0.95
Gas B Mass Fraction		
100	1.12e-04	
200	5.45e-05	1
400	2.74e-05	0.99
800	1.37e-05	1
Gas B Volume Fraction		
100	1.08e-04	
200	5.61e-05	0.95
400	2.88e-05	0.96
800	1.43e-05	1

Table 3.8: Numerical error and convergence rates for the multiphase smooth test. The tests used an *Eulerian grid* with the *predictor on*. (cont)

N	E_N	r_N
Gas A Density		
100	4.31e-04	
200	1.65e-04	1.4
400	7.37e-05	1.2
800	3.45e-05	1.1
Gas A Velocity		
100	4.61e-04	
200	1.81e-04	1.3
400	8.37e-05	1.1
800	4.01e-05	1.1
Gas A Pressure		
100	6.32e-04	
200	2.46e-04	1.4
400	1.10e-04	1.2
800	5.16e-05	1.1
Gas A Energy		
100	4.67e-04	
200	1.91e-04	1.3
400	8.79e-05	1.1
800	4.20e-05	1.1
Gas A Mass Fraction		
100	1.10e-04	
200	4.01e-05	1.5
400	1.77e-05	1.2
800	8.30e-06	1.1
Gas A Volume Fraction		
100	1.53e-04	
200	7.10e-05	1.1
400	3.40e-05	1.1
800	1.63e-05	1.1

Table 3.9: Numerical error and convergence rates for the multiphase smooth test. The tests used an *Eulerian grid* with the *predictor on/no limiter*.

N	E_N	r_N
Gas B Density		
100	3.10e-04	
200	1.26e-04	1.3
400	5.96e-05	1.1
800	2.87e-05	1.1
Gas B Velocity		
100	3.25e-04	
200	1.52e-04	1.1
400	7.60e-05	1
800	3.76e-05	1
Gas B Pressure		
100	4.29e-04	
200	1.75e-04	1.3
400	8.25e-05	1.1
800	3.96e-05	1.1
Gas B Energy		
100	3.35e-04	
200	1.31e-04	1.4
400	6.07e-05	1.1
800	2.89e-05	1.1
Gas B Mass Fraction		
100	1.10e-04	
200	4.01e-05	1.5
400	1.77e-05	1.2
800	8.31e-06	1.1
Gas B Volume Fraction		
100	1.53e-04	
200	7.10e-05	1.1
400	3.40e-05	1.1
800	1.63e-05	1.1

Table 3.9: Numerical error and convergence rates for the multiphase smooth test. The tests used an *Eulerian grid* with the *predictor on/no limiter*. (cont)

3.2.2 Water-Air Shock Tube

Test Description

A classical one-dimensional shock tube with multiple materials is investigated here. The test consists of high-pressure liquid water on the left and low-pressure air on the right. This has become a standard test for multi-fluid algorithms. It has been modeled using interface tracking methods as well as diffuse boundary methods, including single-phase mixture models [86] and multiphase models [20, 22]. The current setup was based on the water-air test problem of Saurel and Abgrall [19].

Test Setup

This test is a one-dimensional simulation of a shock tube with liquid water on the left and air on the right. The current multiphase algorithm is used for this simulation with each fluid being a separate phase. This leads to a diffuse boundary between the materials. No relaxation terms were used in these calculations, i.e. phase interaction due to pressure relaxation, drag, and thermal conduction were neglected. Both fluids are represented by the stiffened-gas equation of state (see §A.1.2). The water is represented by $\gamma = 4.4$ and $P_\infty = 6 \times 10^8$ and the air is given by $\gamma = 1.4$ and $P_\infty = 0$. The spatial domain of the problem is $x \in [0, 1]$ where the left and right states are separated by a discontinuity at position $x = 0.7$. The solutions are compared at the simulation end time $t = 229 \times 10^{-6}$. The initial conditions are tabulated in Table 3.10.

The problem is one dimensional by definition. It is modeled using a three-dimensional mesh with a single zone in the y and z directions and symmetry boundary conditions on the sides, resulting in a pseudo-1D flow in the x direction. The number of zones in the x direction are varied to look at mesh convergence. Outflow boundary conditions are given at the boundaries $x = 0$ and $x = 1$.

This test problem is run on both a fixed and moving mesh in order to compare the two methodologies. For the ALE problem, the mesh motion is predefined and is not dependent on the motion of the material. The motion is defined such that

the velocity of the center node (initially at $x = 0.5$) translates to the right at a constant speed. The center node's translation speed is specified such that it will travel one-quarter of the distance to the end of the domain at the end of the simulation.

Test Theory

An algorithm for an exact Riemann solver for the stiffened-gas equation of state is given by Ivings et al. [87]. This is extended for multiple materials to obtain the exact solution. The resulting solution assumes distinct materials with a discrete material interface. Thus, there is no concept of a separate state for each phase. Therefore, the theoretical values will be plotted against mixture quantities for comparison.

Test Results

The results of the simulations are plotted in Figures 3.67–3.70. The numerical results for the various grid resolutions are given by the colored symbols, while the exact solution is represented by the solid black line. The computations were performed using both the first-order method (predictor off) and the second-order scheme (predictor on). Calculations were done using a fixed Eulerian mesh and a moving ALE mesh.

Similar to the Sod problem, this test consists of a rightward traveling shock, a rarefaction wave on the left, and in the middle there is a contact discontinuity traveling to the right following the shock. This is a very severe problem with large pressure and density differences in the materials.

The first-order scheme (Figure 3.67) exhibits some numerical diffusion in the shock and the contact discontinuity. In fact, there is no distinct separation between the shock and the contact surface. Numerical diffusion has also flattened out the rarefaction wave. The propagation speed of the various waves appears to be correct as the average position of the waves is close to the theoretical position. The volume fraction of the gas is plotted in Figure 3.67(e). This shows that the mixture region of the material interface has been diffused over several cells. This helps explain

the diffusive nature of the contact surface in the mixture quantities. However, even though these calculations exhibit some numerical diffusion of the material interface, it is observed that this is expected and the mixture quantities are still represented with the accuracy one would expect with a first-order method. It also appears that the solutions are converging to the exact solution as the grid is refined.

The numerical results of the ALE water-air test are plotted in Figure 3.69. The results and convergence are comparable to the fixed-grid simulations, although the ALE method seems to better resolve the shock and contact surface. This is due to additional mesh resolution moving into the shock region, and less up-winding resulting in less diffusion.

The results with the predictor on are shown in Figure 3.69 for an Eulerian grid and in Figure 3.70 for an ALE grid. The second-order method shows less diffusion at the discontinuities, especially in the calculation of the volume fraction. There appears to be some slight under/over-shoots in the variables at the ends of the rarefaction and the shock, but these are very minor perturbations.

Test Conclusions

The following conclusions can be drawn from this calculation:

- The diffuse interface treatment in the current algorithm adequately computes the mixture quantities in the multi-material problem.
- The calculations show similar behavior for the water-air problem run on fixed or moving meshes. Thus, it appears that the ALE techniques used here work properly for this test.

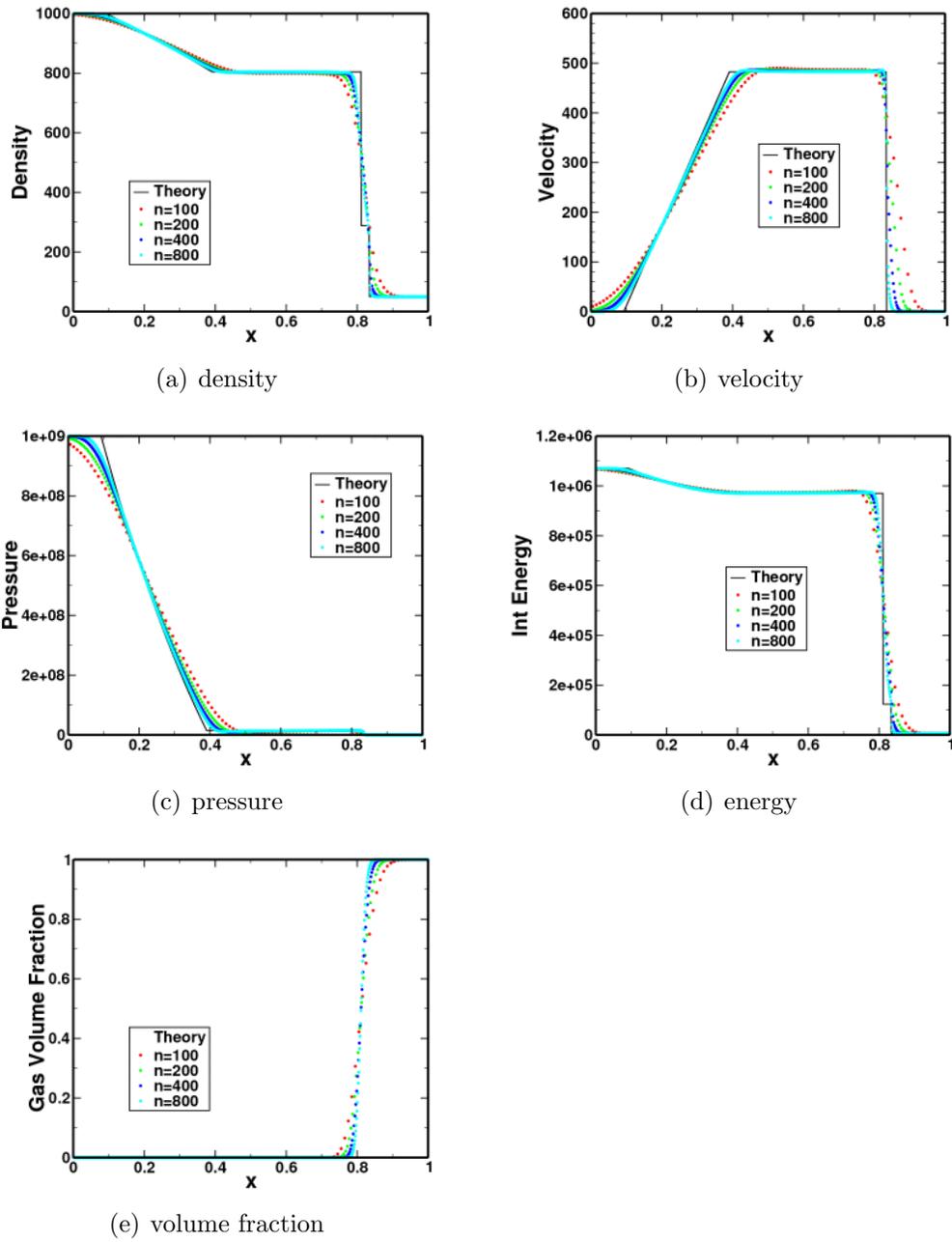


Figure 3.67: Multiphase numerical results for the water-air shock-tube test. The tests used an *Eulerian grid* with the *predictor off*.

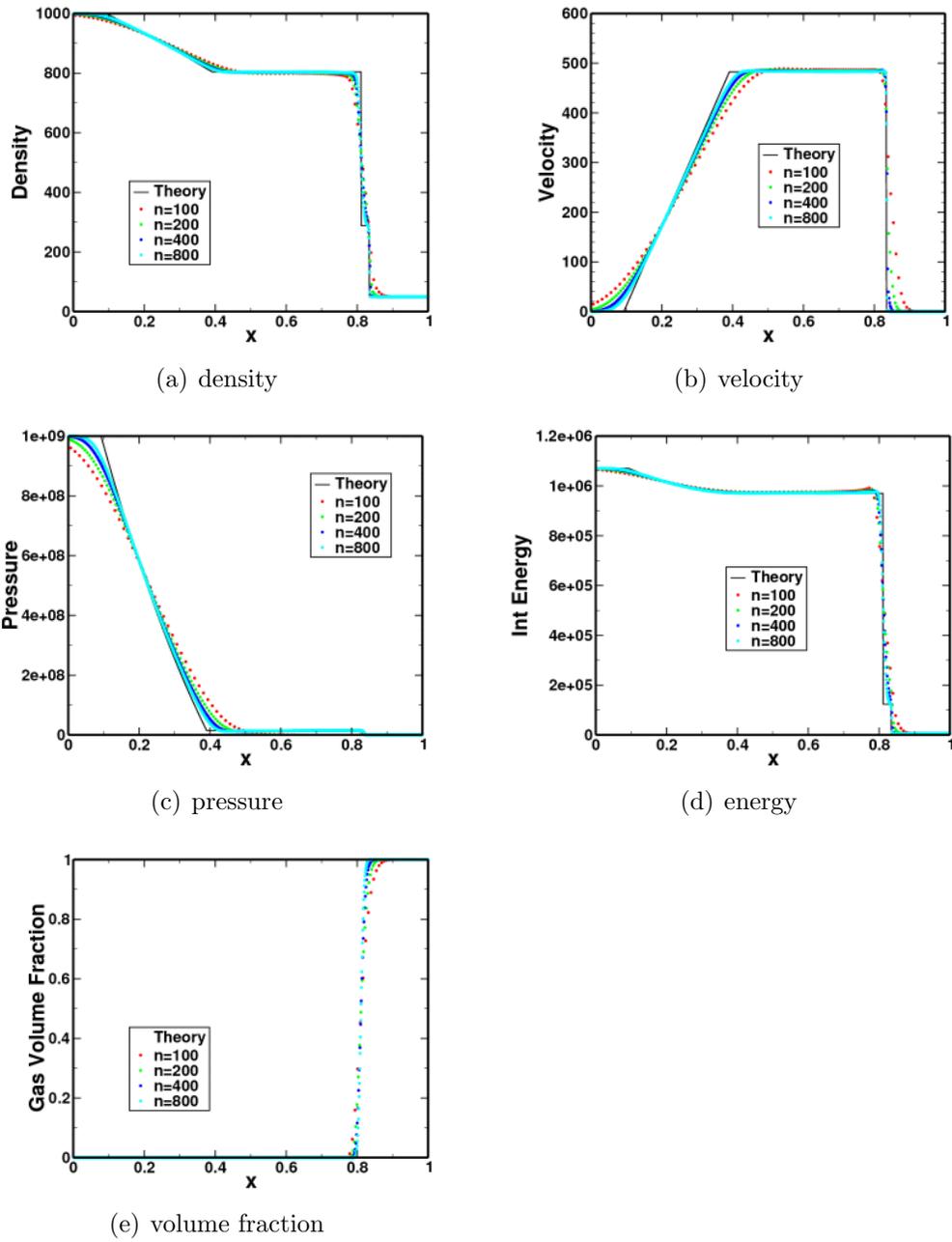


Figure 3.68: Multiphase numerical results for the water-air shock-tube test. The tests used an *ALE grid* with the *predictor off*.

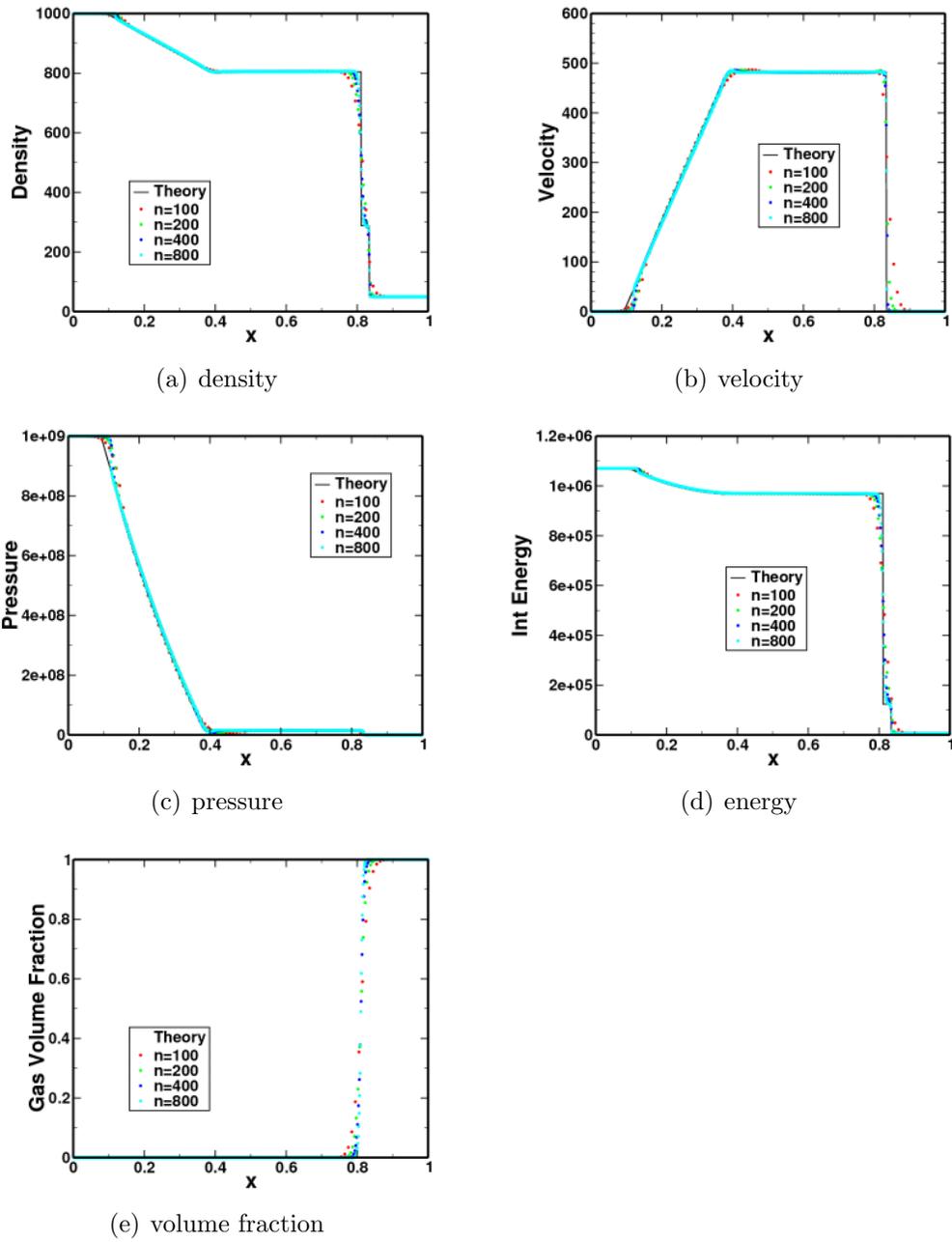


Figure 3.69: Multiphase numerical results for the water-air shock-tube test. The tests used an *Eulerian grid* with the *predictor on*.

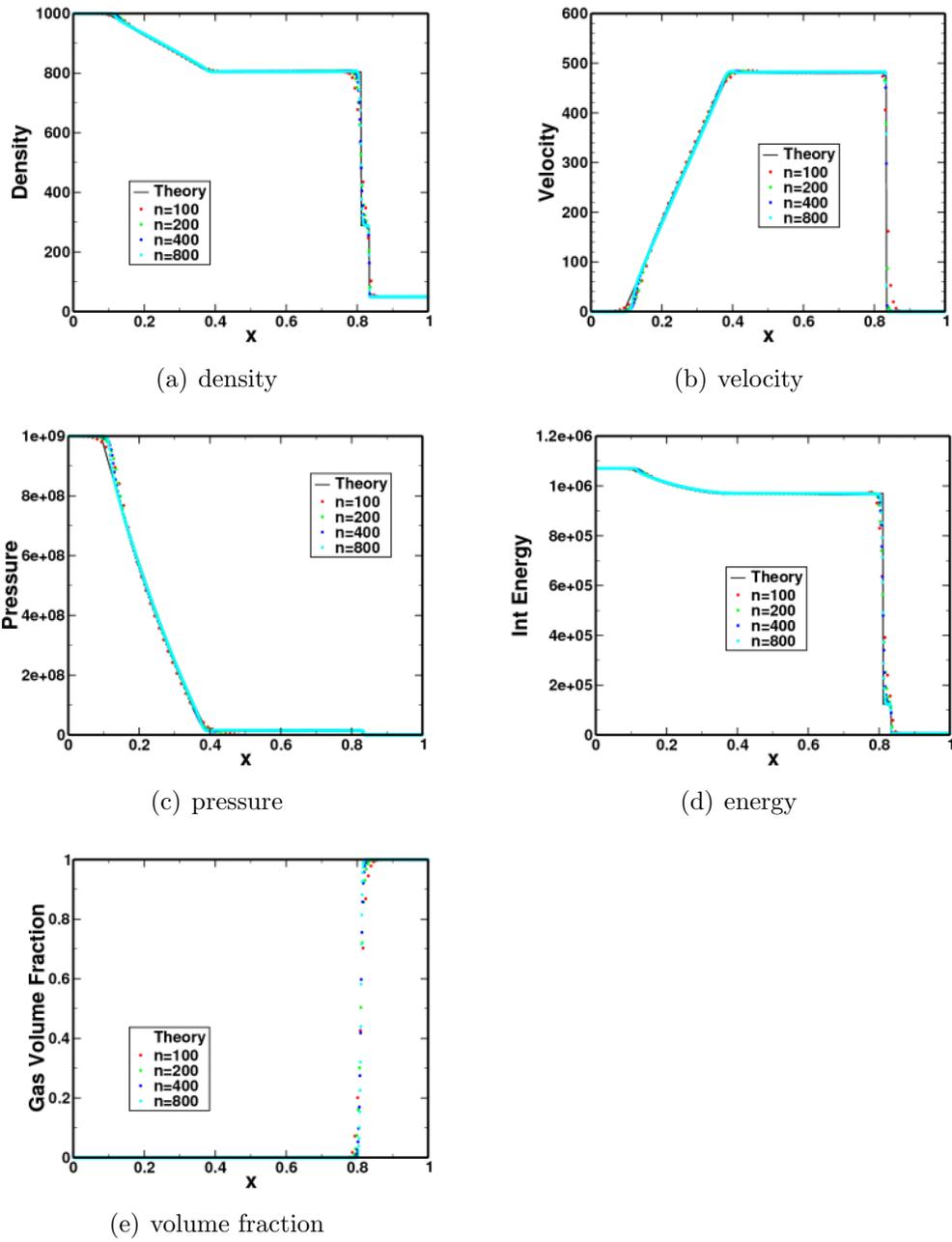


Figure 3.70: Multiphase numerical results for the water-air shock-tube test. The tests used an *ALE grid* with the *predictor on*.

Variable	Value
x_0	0.7
t_0	229×10^{-6}
ρ_L	1000
u_L	0.0
p_L	1×10^9
ρ_R	50
u_R	0.0
p_R	1×10^5

Table 3.10: Initial conditions for the water-air shock-tube problem

3.2.3 Shock-Interface Interaction

Test Description

This is a one-dimensional shock-tube problem with multiple materials. The problem was initially proposed by Banks et al. [88] as a single-phase multi-fluid test. The test consists of a low-density gas on the left and a high-density gas on the right. A shock is initialized in the left material, which propagates to the right through the material interface.

Test Setup

This test is a one-dimensional simulation of a shock tube with a spatial domain of $x \in [0, 1]$. It contains two materials (*gas-a* and *gas-b*) separated by a discontinuity initially located at $x = 0.5$. The interface is translating to the right with a velocity of 0.1 relative to the fixed lab frame. The domain is initially separated into three regions. On the left in Region 1 ($x \in [0.0, 0.1]$) is *gas-a* at a high pressure; in the middle in Region 2 ($x \in (0.1, 0.5]$) is *gas-a* at ambient pressure; on the right in Region 3 ($x \in (0.5, 1.0]$) is *gas-b* at ambient pressure. The high pressure in Region 1 drives a right-traveling shock wave with a shock Mach number of 2.0. This shock wave travels through *gas-a*, eventually interacting with the material interface, resulting in a reflected shock going back into *gas-a* and a transmitted shock in *gas-b*.

The current multiphase algorithm is used for this simulation with *gas-a* and *gas-b* modeled as separate phases. This leads to a diffuse boundary between the materials. No relaxation terms were used in these calculations, i.e. phase interaction due to pressure relaxation, drag, and thermal conduction were neglected. Both fluids are represented with an ideal-gas equation of state. The *gas-a* is represented with $\gamma = 1.35$ and a specific heat $C_v = 2.4$, while *gas-b* has $\gamma = 5.0$ and $C_v = 1.5$. The initial conditions are tabulated in Table 3.11. The solutions are compared at the simulation end time $t = 0.25$.

The problem is one dimensional by definition. It is modeled using an Eulerian three-dimensional mesh with a single zone in the y and z directions and symmetry

boundary conditions on the sides, resulting in a pseudo-1D flow in the x direction. The number of zones in the x direction are varied to look at mesh convergence. Outflow boundary conditions are given at the boundaries $x = 0$ and $x = 1$.

Test Theory

The initial state of the gas behind the shock (Region 1) can be computed using the ideal-gas normal-shock relations described in Zucker [89]. After the shock impacts the material interface, the resulting state can be determined by solving suitable Riemann problems as discussed in Toro [53]. The flow properties are determined at every spatial location using the method of characteristics, as discussed in Zel'dovich and Raizer [90].

Test Results

The results of the simulations are plotted in Figures 3.71–3.72. The numerical results for the various grid resolutions are given by the colored symbols, while the exact solution is represented by the solid black line. The computations were performed using both the first-order method (predictor off) and the second-order scheme (predictor on). Calculations were done using a fixed Eulerian mesh.

The results of this test consist of a leftward traveling reflected shock, a rightward traveling transmitted shock, and a contact discontinuity in the middle traveling to the right. The first-order approximation (Figure 3.71) exhibits some numerical diffusion in the discontinuities. The simulations with the predictor on (Figure 3.72) resolve the discontinuities better. The propagation speed of the various waves appears to be correct in all simulations as the average position of the waves is close to the theoretical position. It is observed that there exists a small perturbation in the results to the far left. This was caused by start-up errors and was observed by Banks [88] as well. It is noted that the undershoot in density observed by Banks is not present with the current method. The volume fraction plots show that the mixture region of the material interface has been diffused over several cells. Despite the diffuse interface, it is observed that the mixture quantities are still represented accurately. It also appears that the solutions are converging to

the exact solution as the grid is refined.

Test Conclusions

The following conclusions can be drawn from this calculation:

- The diffuse-interface treatment in the current algorithm adequately computes the mixture quantities in the multi-material problem.
- The discrete equation method does not exhibit the density undershoot discussed for the uncorrected scheme in Banks et al. [88].

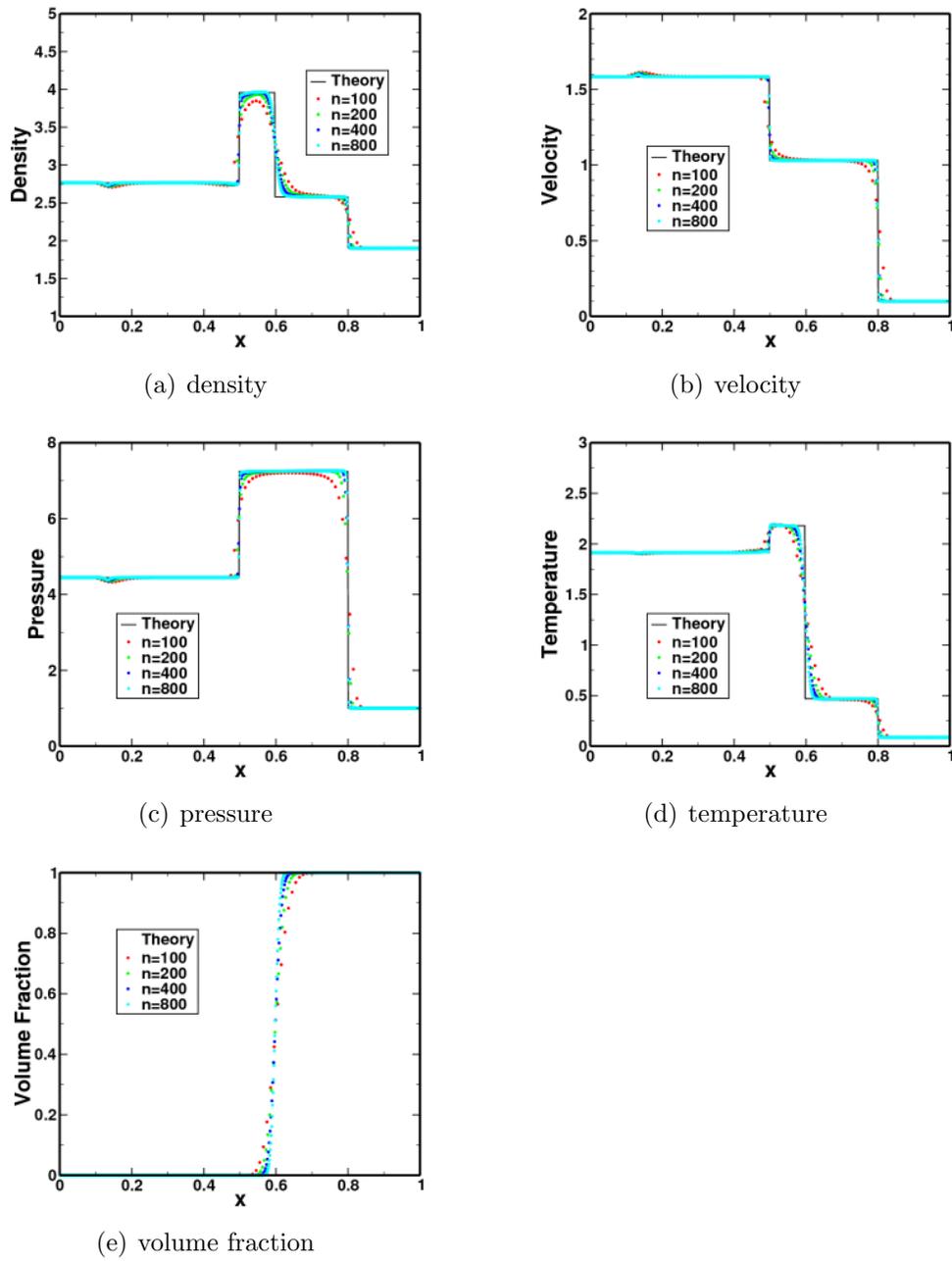


Figure 3.71: Multiphase numerical results for the shock-interface test. The tests used an *Eulerian grid* with the *predictor off*.

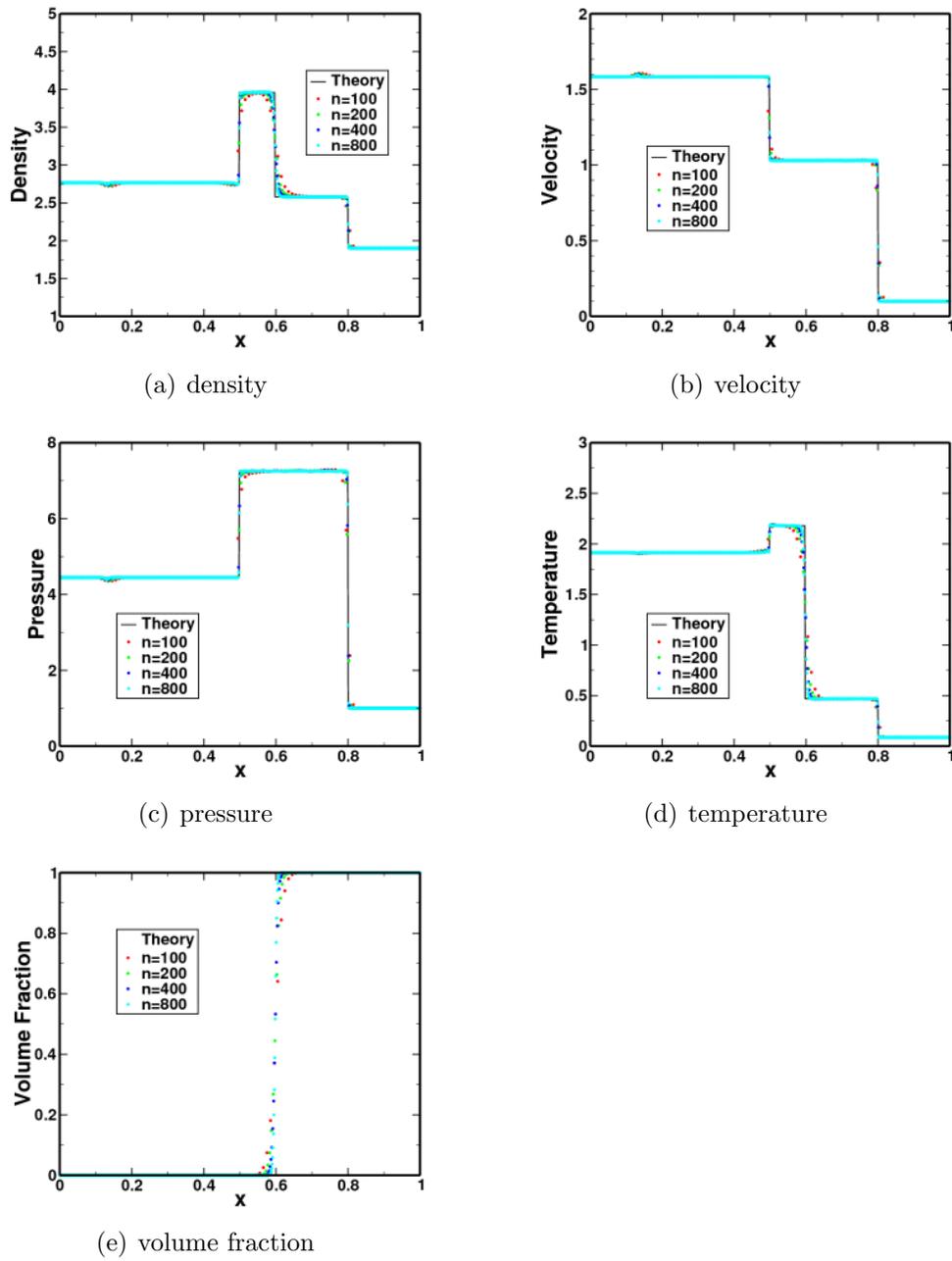


Figure 3.72: Multiphase numerical results for the shock-interface test. The tests used an *Eulerian grid* with the *predictor on*.

Variable	Region 1	Region 2	Region 3
ρ	2.7647	1.0	1.9
u	1.5833	0.1	0.1
p	4.4468	1.0	1.0

Table 3.11: Initial conditions for the shock-interface test problem.

3.2.4 Multiphase Temperature Relaxation

Test Description

This is a test of the multiphase temperature-relaxation (heat-transfer) model. It is a single-element problem with a gas and solid phase. The only interaction between the phases occurs due to temperature relaxation.

Test Setup

This problem consists of a single gas phase (a) and a single solid phase (b). The geometry is a single element with outflow boundary conditions enforced on all external faces. Thus, there are no flow gradients possible and the hydrodynamic equations are trivially satisfied. Initially, the phases are at different temperatures, resulting in conductive heat transfer. The gas phase is initially at temperature $\bar{T}_a = 1$ with a mass fraction of $x_a = 0.8$, and the solid phase is initially at temperature $\bar{T}_b = 2$ with a mass fraction of $x_b = 0.2$. The specific heats of both phases are one ($C_{va} = C_{vb} = 1.0$). The thermal relaxation time is given by a constant value of $\tau = 0.5$.

Test Theory

Crowe et al. [4] provides an analytical expression for the thermal response of a discrete phase (d) embedded within a continuous phase (c)

$$\frac{dT_d}{dt} = \frac{1}{\tau} (T_c - T_d)$$

where the thermal response time is given by τ . This expression has been extended to multiple phases in a closed system in §2.1.14.2. For the two-phase problem of interest here, the governing equations reduce to

$$\begin{aligned} \frac{dT_a}{dt} &= -\frac{c}{\tau} (T_a - T_b) \\ \frac{dT_b}{dt} &= \frac{1}{\tau} (T_a - T_b) \end{aligned}$$

where

$$c = \frac{x_b C_{vb}}{x_a C_{va}}$$

is a constant. From the conditions given above, $c = 0.25$. By assuming an exponential form and utilizing the initial conditions, the solution to this system is found to be

$$\begin{aligned} T_a &= \bar{\bar{T}}_a + \frac{c}{c+1} (\bar{\bar{T}}_a - \bar{\bar{T}}_b) (e^{-\beta t} - 1) \\ T_b &= \bar{\bar{T}}_b - \frac{1}{c+1} (\bar{\bar{T}}_a - \bar{\bar{T}}_b) (e^{-\beta t} - 1) \end{aligned}$$

where $\beta = \frac{c+1}{\tau} = 2.5$.

Test Results

The calculated temperatures show good agreement with the theoretical solution. Calculations were performed using the backward-Euler and the analytical integration methods. The simulations were all run over a time interval of one using constant time-steps. In order to show the temporal convergence of the schemes, the simulations used three different time-step sizes, such that 10, 20, and 40 time-steps were used for the simulations.

The backward-Euler results are shown in Figure 3.73. As the temporal refinement is improved, the method appears to be converging toward the theoretical solution. It is expected that the solution rate should be first order for the backward-Euler method.

Results from the analytical integration method are shown in Figure 3.74. As expected, the results are nearly identical to the theoretical values. This is because the analytical method solves the theoretical equations directly. Thus, for this problem there are no discretization errors associated with this method. Any errors in these calculations are the result of round-off errors.

Test Error Quantification and Convergence

There is a theoretical solution to this problem. Therefore, the discretization error for the solution can be quantified. Since there is no spatial variation in the solution to this problem, spatial-discretization error does not contribute to the total error of the calculation and the time-discretization error should dominate.

The error (E_N) is determined for time-step refinement level N using the

methods of §E. The calculated error has been computed for the temperature of both phases separately. These errors are tabulated in Table 3.12 for the backward-Euler method and Table 3.13 for the calculations using the analytic integration method. The convergence rates (r_N) are also tabulated. As expected, the convergence rates for the backward-Euler method are around one, indicating first-order convergence. On the other hand, the error for the analytic solution increases as the time-step is decreased, thus the solution does not converge with refinement. This is because the code is computing an exact solution analytically. Therefore, there is no time-discretization error in these calculations and the error in the calculation is dominated by round-off errors. This results in error norms which are many orders of magnitude smaller for the analytic method compared to the backward Euler.

Test Conclusions

The following conclusions can be drawn from this calculation:

- The backward-Euler temperature-relaxation method provides a reasonable solution for this problem. It converges toward the theoretical solution as expected for a first-order method.
- The analytical temperature-relaxation method provides an exact solution for this problem.

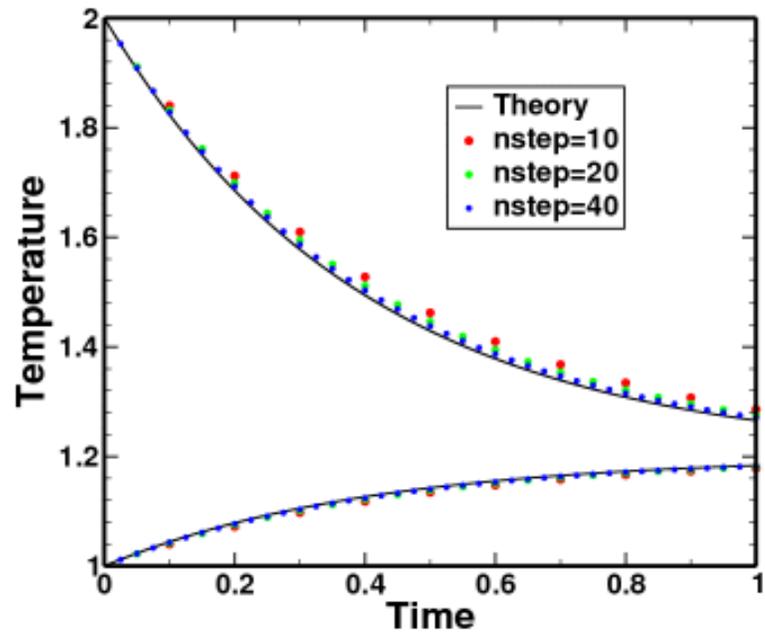


Figure 3.73: Multiphase numerical results for the temperature-relaxation test. Phase temperatures were computed using the backward-Euler method. The higher temperature curves represent the solid (phase *b*) and the lower temperature curves represent the gas (phase *a*).

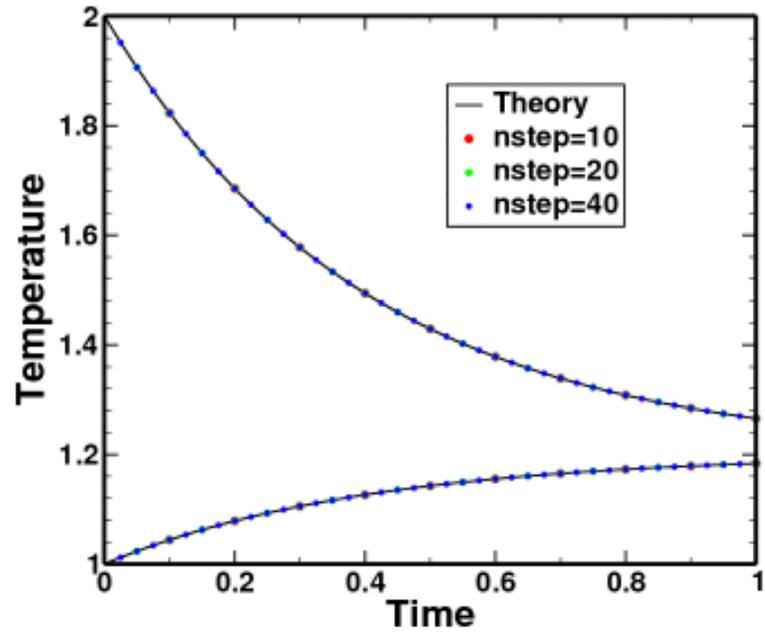


Figure 3.74: Multiphase numerical results for the temperature-relaxation test. Phase temperatures were computed using the analytical method. The higher temperature curves represent the solid (phase *b*) and the lower temperature curves represent the gas (phase *a*).

N	E_N	r_N
Phase a Temperature		
10	6.90e-03	
20	3.57e-03	0.95
40	1.82e-03	0.97
Phase b Temperature		
10	2.76e-02	
20	1.43e-02	0.95
40	7.26e-03	0.97

Table 3.12: Numerical error and convergence rates for the temperature-relaxation test. Phase temperatures were computed using the backward-Euler method.

N	E_N	r_N
Phase a Temperature		
10	5.80e-13	
20	1.18e-12	-1
40	2.36e-12	-1
Phase b Temperature		
10	2.33e-12	
20	4.71e-12	-1
40	9.44e-12	-1

Table 3.13: Numerical error and convergence rates for the temperature-relaxation test. Phase temperatures were computed using the analytical method.

3.2.5 Multiphase Nusselt Number

Test Description

This tests the multiphase temperature-relaxation (heat-transfer) model and Nusselt-number formulations. It is a single-element problem with a gas and solid phase. The only interaction between the phases occurs due to temperature relaxation.

Test Setup

This problem consists of a single gas phase (a) and a single solid phase (b). The geometry is a single element with outflow boundary conditions enforced on all external faces. Thus, there are no flow gradients possible and the hydrodynamic equations are trivially satisfied. Initially, the phases are at different temperatures, resulting in conductive heat transfer. The gas phase is initially at temperature $\bar{T}_a = 1$ and the solid phase is initially at temperature $\bar{T}_b = 2$. The specific heats of both phases are one ($C_{va} = C_{vb} = 1.0$). The specific heats, conductivity, and viscosity are all taken as constant in this test.

The problem is run with a variety of empirical relations for the Nusselt number in order to verify that they are all implemented correctly. The functional forms for the Nusselt number are discussed in §C. Since many of the relations for the Nusselt number depend on Reynolds number, a velocity difference of one is given to the materials. Since the velocity and all other parameters are constant, the Reynolds number is also constant with a value of 2500. The tests are repeated for three different porosity levels with gas mass fractions of 95%, 70%, and 20%.

Test Theory

The analytical expression for the temperature evolution of a multiphase mixture from heat transfer between the phases is discussed in §2.1.14.2. For the two-phase problem of interest here, the governing equations reduce to

$$\begin{aligned}\frac{dT_a}{dt} &= -\frac{c}{\tau} (T_a - T_b) \\ \frac{dT_b}{dt} &= \frac{1}{\tau} (T_a - T_b)\end{aligned}$$

where $c = \frac{(x_b C_{vb})}{(x_a C_{va})}$. Since the mass fractions and specific heats do not change during the simulation, c is a constant in this problem.

In §3.2.4 the relaxation time, τ , was constant. In the current problem however, τ is no longer a constant, but is determined by the definition

$$\tau = \frac{1}{12} \frac{\rho_b C_{vb} d_b^2}{k_a} \frac{2}{Nu}$$

where the Nusselt number (Nu) itself depends on flow properties of the fluid surrounding the particle, such as Reynolds number, Prandtl number, and porosity. As such, τ could potentially vary with time and the analytic expression used for comparison in §3.2.6 is not valid. Therefore, rather than obtaining an analytic expression for comparison, an accurate numerical solution of the above system of ODE's is used. The numerical solution is calculated using the high-order Adams method in SciPy [91]. Since the method used to integrate the multiphase heat transfer terms is expected to be at most first-order accurate in time, the numerical solution is an adequate representation of the *exact* theoretical solution. Note that although τ is not constant in general, for this particular test problem it is constant since none of the models used are temperature dependent.

Test Results

The calculated temperatures show good agreement with the high-order numerical solution. Calculations were performed using the backward-Euler and the analytical integration methods. The simulations were all run over a time interval of one using constant time-steps. In order to show the temporal convergence of the schemes, the simulations used three different time-step sizes, such that 10, 20, and 40 time-steps were used for the simulations. The tests were repeated for three different porosity levels to see the effect of volume fraction.

The results for each Nusselt-number law are plotted in the following figures. Each figure contains six subplots which are organized in a table with two columns and three rows. Each subplot compares the high-order solution (solid line) to the current algorithm's solution (colored symbols) for three different time-step sizes. The left column contains the results using the backward-Euler time integration;

while the right column uses the analytic time integration. Each row plots the data from a specific porosity level: the top row contains data for a gas mass fraction of 95%, the middle row 70%, and the bottom row 20%. The Nusselt-number models tested are the fixed relaxation constant (Figure 3.75), constant Nusselt number (Figure 3.76), and the Ranz-Marshall (Figure 3.77).

As would be expected, the results for a fixed relaxation constant are similar to those obtained in §3.2.4. The backward-Euler method converges to the theoretical solution as the time-step size is reduced. The analytic integration method reproduced the exact solution for all time-refinement levels.

For the current test configuration, the Nusselt-number models result in a constant relaxation time. Thus, the results for all Nusselt-number models exhibit similar behavior as the fixed relaxation time option.

Test Error Quantification and Convergence

The discretization error for the solution can be quantified using the high-accuracy numerical solution. Since there is no spatial variation in the solution to this problem, spatial-discretization error does not contribute to the total error of the calculation and the time-discretization error should dominate.

The error (E_N) is determined for time-step refinement level N using the methods of §E. The calculated error has been computed for the temperature of both phases separately. These errors are tabulated in Tables 3.14–3.16. The convergence rates (r_N) are also tabulated. These tables are organized in rows and columns similar to the plots discussed earlier in this section.

It is expected that the methods should produce at least first-order convergence in each of the tests. Since all Nusselt-number models result in a constant relaxation time, the analytic integration method essentially reproduces the exact solution, and therefore the errors are dominated by round-off and the solutions do not exhibit convergence. In fact, for the current tests, the analytic method is expected to give more accurate results than the high-accuracy numerical solution used for the *exact* solution. The resulting convergence rates for the backward-Euler

integration method appear to indicate first-order convergence, although in some cases the solutions do not appear to have reached the asymptotic regime and so the convergence is less than first order. It is expected that running additional cases with smaller time-step would exhibit first-order behavior. This was verified by running the tests with 1000, 2000, and 4000 time-step where first-order convergence was observed in all cases.

Test Conclusions

The following conclusions can be drawn from this calculation:

- The numerical results seem to adequately predict the theoretical solutions for all Nusselt-number laws. In general, the solutions appear to exhibit first-order convergence behavior as expected.

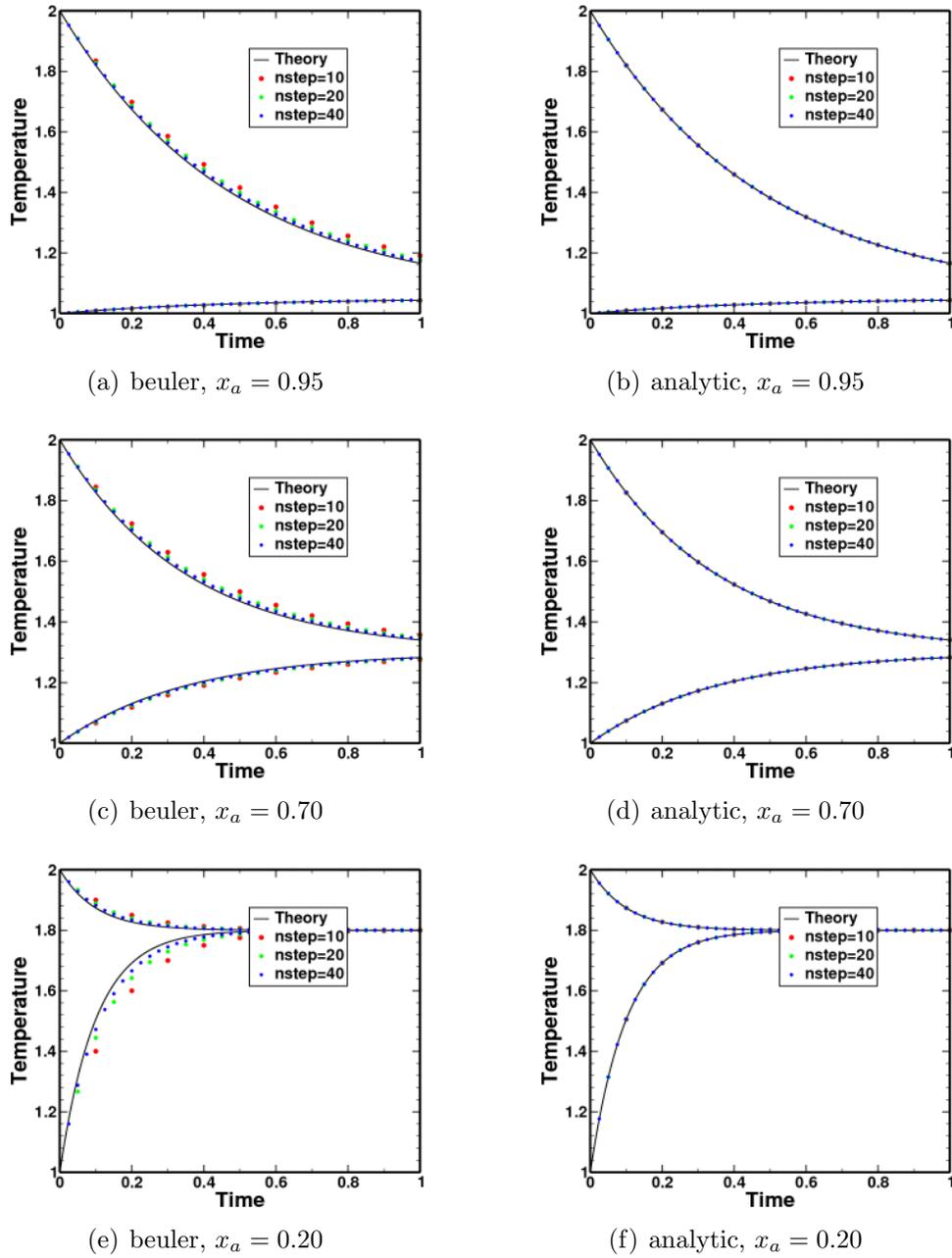


Figure 3.75: Multiphase numerical results for the Nusselt-number tests. The Nusselt-number law used a *fixed relaxation constant*. The higher temperature curves represent the solid (phase *b*) and the lower temperature curves represent the gas (phase *a*).

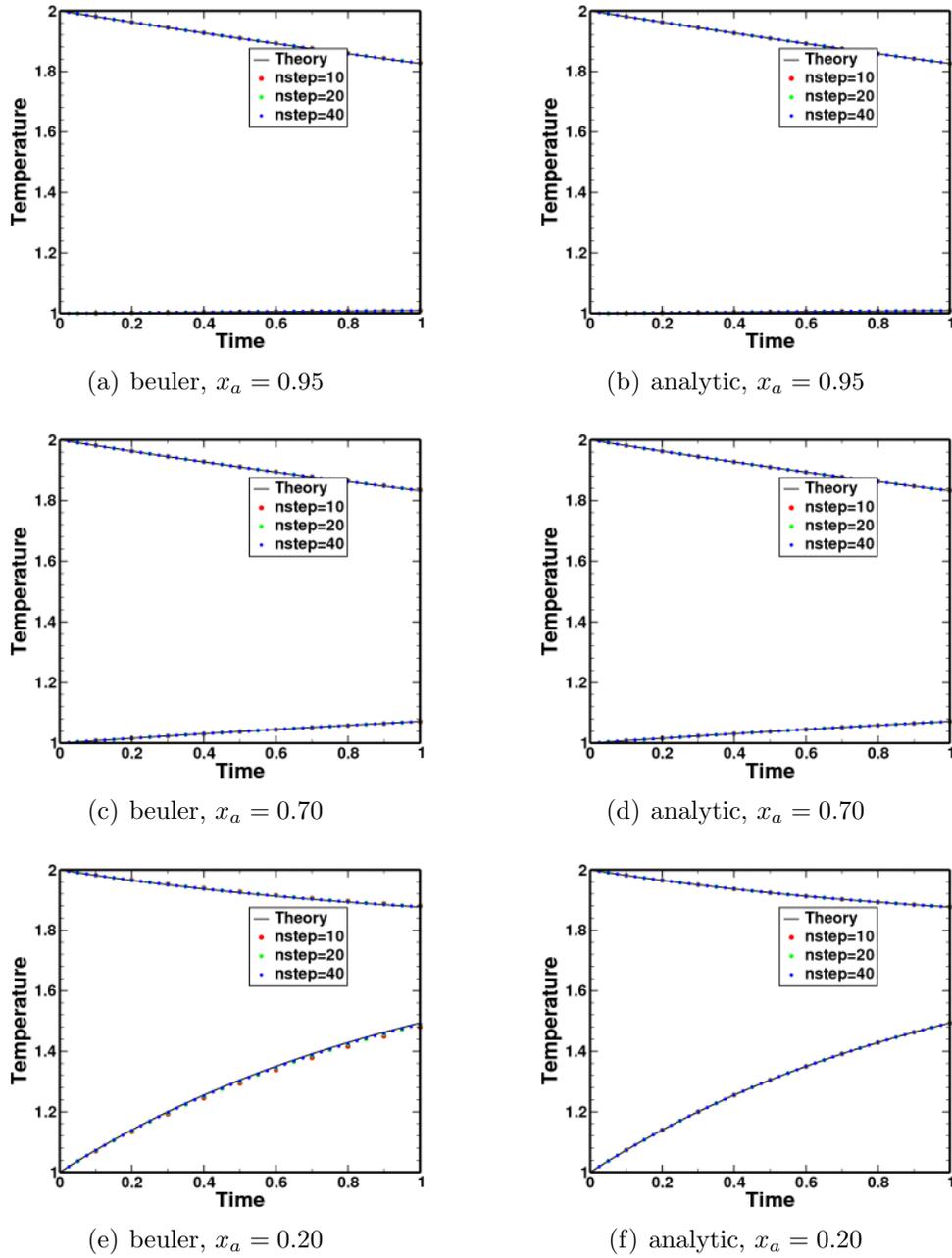


Figure 3.76: Multiphase numerical results for the Nusselt-number tests. The Nusselt-number law used a *constant Nusselt number*. The higher temperature curves represent the solid (phase *b*) and the lower temperature curves represent the gas (phase *a*).

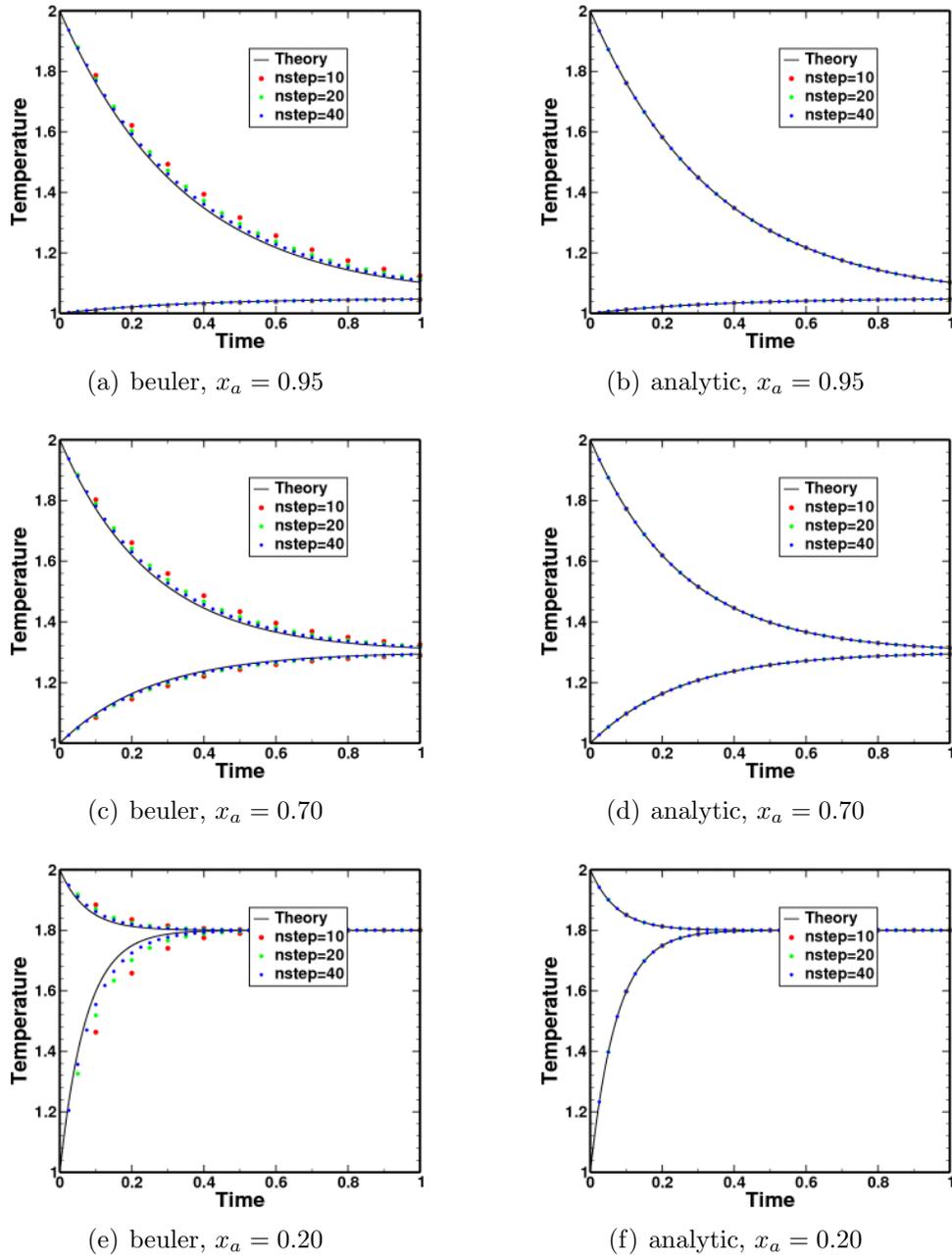


Figure 3.77: Multiphase numerical results for the Nusselt-number tests. The Nusselt-number law used a *Ranz-Marshall correlation*. The higher temperature curves represent the solid (phase *b*) and the lower temperature curves represent the gas (phase *a*).

N	E_N	r_N
Phase a Temperature		
10	1.52e-03	
20	7.83e-04	0.96
40	3.97e-04	0.98
Phase b Temperature		
10	2.90e-02	
20	1.49e-02	0.96
40	7.55e-03	0.98

(a) beuler, $x_a = 0.95$

N	E_N	r_N
Phase a Temperature		
10	1.36e-09	
20	1.13e-09	0.27
40	1.12e-09	0.0043
Phase b Temperature		
10	2.59e-08	
20	2.14e-08	0.27
40	2.13e-08	0.0043

(b) analytic, $x_a = 0.95$

N	E_N	r_N
Phase a Temperature		
10	1.13e-02	
20	5.85e-03	0.95
40	2.98e-03	0.97
Phase b Temperature		
10	2.63e-02	
20	1.36e-02	0.95
40	6.96e-03	0.97

(c) beuler, $x_a = 0.70$

N	E_N	r_N
Phase a Temperature		
10	1.11e-08	
20	9.66e-09	0.2
40	9.54e-09	0.019
Phase b Temperature		
10	2.59e-08	
20	2.25e-08	0.2
40	2.23e-08	0.019

(d) analytic, $x_a = 0.70$

N	E_N	r_N
Phase a Temperature		
10	5.00e-02	
20	2.79e-02	0.84
40	1.48e-02	0.91
Phase b Temperature		
10	1.25e-02	
20	6.97e-03	0.84
40	3.70e-03	0.91

(e) beuler, $x_a = 0.20$

N	E_N	r_N
Phase a Temperature		
10	9.81e-09	
20	9.59e-09	0.033
40	1.02e-08	-0.096
Phase b Temperature		
10	2.45e-09	
20	2.40e-09	0.033
40	2.56e-09	-0.096

(f) analytic, $x_a = 0.20$

Table 3.14: Numerical error and convergence rates for the Nusselt-number tests. The Nusselt-number law used a *fixed relaxation constant*.

N	E_N	r_N
Phase a Temperature		
10	5.35e-05	
20	2.61e-05	1
40	1.29e-05	1
Phase b Temperature		
10	1.02e-03	
20	4.95e-04	1
40	2.44e-04	1

(a) beuler, $x_a = 0.95$

N	E_N	r_N
Phase a Temperature		
10	1.93e-11	
20	1.55e-11	0.32
40	1.36e-11	0.18
Phase b Temperature		
10	3.66e-10	
20	2.94e-10	0.32
40	2.59e-10	0.18

(b) analytic, $x_a = 0.95$

N	E_N	r_N
Phase a Temperature		
10	5.57e-04	
20	2.72e-04	1
40	1.35e-04	1
Phase b Temperature		
10	1.30e-03	
20	6.35e-04	1
40	3.14e-04	1

(c) beuler, $x_a = 0.70$

N	E_N	r_N
Phase a Temperature		
10	3.78e-10	
20	3.05e-10	0.31
40	2.69e-10	0.18
Phase b Temperature		
10	8.82e-10	
20	7.12e-10	0.31
40	6.28e-10	0.18

(d) analytic, $x_a = 0.70$

N	E_N	r_N
Phase a Temperature		
10	1.07e-02	
20	5.37e-03	1
40	2.69e-03	1
Phase b Temperature		
10	2.68e-03	
20	1.34e-03	1
40	6.72e-04	1

(e) beuler, $x_a = 0.20$

N	E_N	r_N
Phase a Temperature		
10	2.48e-08	
20	2.36e-08	0.072
40	2.30e-08	0.034
Phase b Temperature		
10	6.20e-09	
20	5.89e-09	0.072
40	5.76e-09	0.034

(f) analytic, $x_a = 0.20$

Table 3.15: Numerical error and convergence rates for the Nusselt-number tests. The Nusselt-number law used a *constant Nusselt number*.

N	E_N	r_N
Phase a Temperature		
10	1.89e-03	
20	9.82e-04	0.95
40	5.01e-04	0.97
Phase b Temperature		
10	3.59e-02	
20	1.87e-02	0.95
40	9.52e-03	0.97

(a) beuler, $x_a = 0.95$

N	E_N	r_N
Phase a Temperature		
10	1.52e-09	
20	1.21e-09	0.32
40	1.12e-09	0.12
Phase b Temperature		
10	2.88e-08	
20	2.30e-08	0.32
40	2.12e-08	0.12

(b) analytic, $x_a = 0.95$

N	E_N	r_N
Phase a Temperature		
10	1.33e-02	
20	6.98e-03	0.93
40	3.58e-03	0.96
Phase b Temperature		
10	3.10e-02	
20	1.63e-02	0.93
40	8.36e-03	0.96

(c) beuler, $x_a = 0.70$

N	E_N	r_N
Phase a Temperature		
10	7.95e-09	
20	9.02e-09	-0.18
40	8.22e-09	0.13
Phase b Temperature		
10	1.86e-08	
20	2.10e-08	-0.18
40	1.92e-08	0.13

(d) analytic, $x_a = 0.70$

N	E_N	r_N
Phase a Temperature		
10	5.40e-02	
20	3.14e-02	0.78
40	1.69e-02	0.89
Phase b Temperature		
10	1.35e-02	
20	7.84e-03	0.78
40	4.24e-03	0.89

(e) beuler, $x_a = 0.20$

N	E_N	r_N
Phase a Temperature		
10	7.66e-09	
20	7.02e-09	0.12
40	8.07e-09	-0.2
Phase b Temperature		
10	1.91e-09	
20	1.76e-09	0.12
40	2.02e-09	-0.2

(f) analytic, $x_a = 0.20$

Table 3.16: Numerical error and convergence rates for the Nusselt-number tests. The Nusselt-number law used a *Ranz-Marshall correlation*.

3.2.6 Multiphase Velocity Relaxation

Test Description

This is a test of the multiphase velocity-relaxation (drag) model. It is a single-element problem with a gas and solid phase. The only interaction between the phases occurs due to velocity relaxation.

Test Setup

This problem consists of a single gas phase (a) and a single solid phase (b). The geometry is a single element with outflow boundary conditions enforced on all external faces. Thus, there are no flow gradients possible and the hydrodynamic equations are trivially satisfied. Initially, the phases are at different velocities, resulting in momentum exchange through drag. The gas phase is initially at velocity $\bar{U}_a = 1$ with a mass fraction of $x_a = 0.8$, and the solid phase is initially at velocity $\bar{U}_b = 2$ with a mass fraction of $x_b = 0.2$. The velocity relaxation time is given by a constant value of $\tau = 0.5$.

Test Theory

Crowe et al. [4] provides an analytical expression for the velocity evolution of a discrete phase (d) embedded within a continuous phase (c)

$$\frac{d\vec{u}_d}{dt} = \frac{1}{\tau} (\vec{u}_c - \vec{u}_d)$$

where the velocity response time is given by τ . This expression has been extended to multiple phases in a closed system in §2.1.14.3. For the two-phase problem of interest here, the governing equations reduce to

$$\begin{aligned} \frac{d\vec{u}_a}{dt} &= -\frac{c}{\tau} (\vec{u}_a - \vec{u}_b) \\ \frac{d\vec{u}_b}{dt} &= \frac{1}{\tau} (\vec{u}_a - \vec{u}_b) \end{aligned}$$

where

$$c = \frac{x_b}{x_a}$$

is a constant. From the conditions given above, $c = 0.25$. By assuming an exponential form and utilizing the initial conditions, the solution to this system is

found to be

$$\begin{aligned}\vec{u}_a &= \bar{U}_a + \frac{c}{c+1} (\bar{U}_a - \bar{U}_b) (e^{-\beta t} - 1) \\ \vec{u}_b &= \bar{U}_b - \frac{1}{c+1} (\bar{U}_a - \bar{U}_b) (e^{-\beta t} - 1)\end{aligned}$$

where $\beta = \frac{c+1}{\tau} = 2.5$.

Test Results

The calculated velocities show good agreement with the theoretical solution. Calculations were performed using the backward-Euler and the analytical integration methods. The simulations were all run over a time interval of one using constant time-steps. In order to show the temporal convergence of the schemes, the simulations used three different time-step sizes, such that 10, 20, and 40 time-steps were used for the simulations.

The backward-Euler results are shown in Figure 3.78. As the temporal refinement is improved, the method appears to be converging toward the theoretical solution. It is expected that the solution rate should be first order for the backward-Euler method.

Results from the analytical integration method are shown in Figure 3.79. As expected, the results are nearly identical to the theoretical values. This is because the analytical method solves the theoretical equations directly. Thus, for this problem there are no discretization errors associated with this method. Any errors in these calculations are the result of round-off errors.

Test Error Quantification and Convergence

There is a theoretical solution to this problem. Therefore, the discretization error for the solution can be quantified. Since there is no spatial variation in the solution to this problem, spatial-discretization error does not contribute to the total error of the calculation and the time-discretization error should dominate.

The error (E_N) is determined for time-step refinement level N using the methods of §E. The calculated error has been computed for the velocity of both phases separately. These errors are tabulated in Table 3.17 for the backward-Euler

method and Table 3.18 for the calculations using the analytic integration method. The convergence rates (r_N) are also tabulated. As expected, the convergence rates for the backward-Euler method are around one, indicating first-order convergence. On the other hand, the error for the analytic solution increases as the time-step is decreased, thus the solution does not converge with refinement. This is because the code is computing an exact solution analytically. Therefore, there is no time-discretization error in these calculations and the error in the calculation is dominated by round-off errors. This results in error norms which are many orders of magnitude smaller for the analytic method compared to the backward Euler.

Test Conclusions

The following conclusions can be drawn from this calculation:

- The backward-Euler velocity-relaxation method provides a reasonable solution for this problem. It converges toward the theoretical solution as expected for a first-order method.
- The analytical velocity-relaxation method provides an exact solution for this problem.

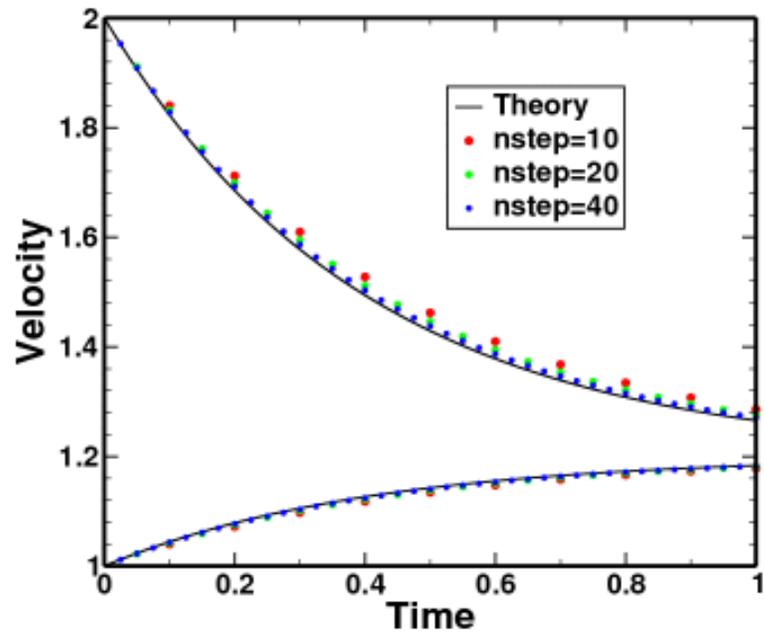


Figure 3.78: Multiphase numerical results for the velocity-relaxation test. Phase velocities were computed using the backward-Euler method. The higher velocity curves represent the solid (phase b) and the lower velocity curves represent the gas (phase a).

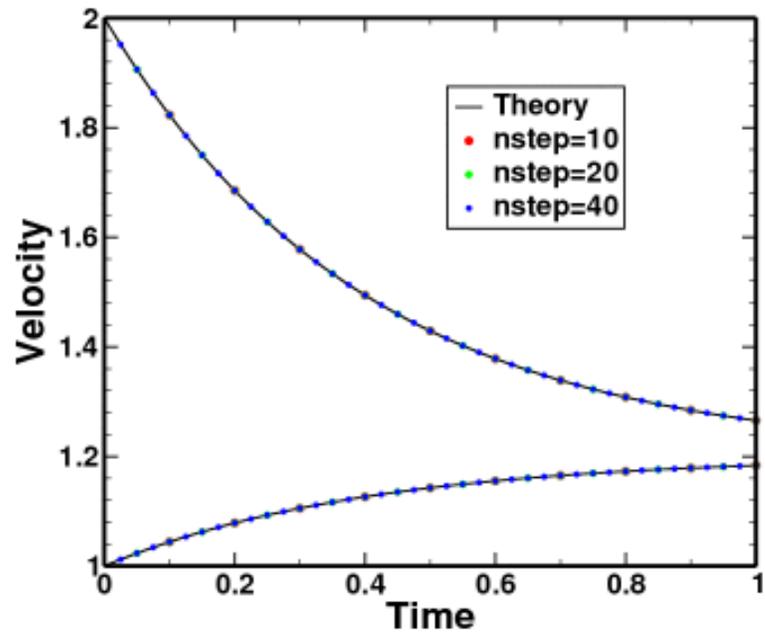


Figure 3.79: Multiphase numerical results for the velocity-relaxation test. Phase velocities were computed using the analytical method. The higher velocity curves represent the solid (phase b) and the lower velocity curves represent the gas (phase a).

N	E_N	r_N
Phase a Velocity		
10	6.90e-03	
20	3.57e-03	0.95
40	1.82e-03	0.97
Phase b Velocity		
10	2.76e-02	
20	1.43e-02	0.95
40	7.26e-03	0.97

Table 3.17: Numerical error and convergence rates for the velocity-relaxation test. Phase velocities were computed using the backward-Euler method.

N	E_N	r_N
Phase a Velocity		
10	5.80e-13	
20	1.18e-12	-1
40	2.36e-12	-1
Phase b Velocity		
10	2.33e-12	
20	4.71e-12	-1
40	9.44e-12	-1

Table 3.18: Numerical error and convergence rates for the velocity-relaxation test. Phase velocities were computed using the analytical method.

3.2.7 Multiphase Drag Coefficient

Test Description

This tests the multiphase velocity-relaxation (drag) model and drag coefficient formulations. It is a single-element problem with a gas and solid phase. The only interaction between the phases occurs due to velocity relaxation.

Test Setup

This problem consists of a single gas phase (a) and a single solid phase (b). The geometry is a single element with outflow boundary conditions enforced on all external faces. Thus, there are no flow gradients possible and the hydrodynamic equations are trivially satisfied. Initially, the phases are at different velocities, resulting in momentum exchange through drag. The gas phase is initially at velocity $\bar{\bar{U}}_a = 1$ and the solid phase is initially at velocity $\bar{\bar{U}}_b = 2$.

The problem is run with a variety of empirical relations for the drag coefficient in order to verify that they are all implemented correctly. The functional forms for the drag coefficient are discussed in §B. Since many of the relations for the drag coefficient depend on Reynolds number, it is desirable for the tests to exercise all relevant regimes. To do this, the problem is initialized such that the initial Reynolds number is 2500. As the simulation advances, drag causes the relative velocity between the phases to decrease, thus reducing the Reynolds number. The variation in Reynolds number between zero and 2500 is adequate to exercise all discontinuous switches in the drag relations. Since many of the drag-coefficient forms also depend on the porosity of the material (volume fraction), the tests are repeated for three different porosity levels with gas mass fractions of 95%, 70%, and 20%.

Test Theory

The analytical expression for the velocity evolution of a multiphase mixture due to drag is discussed in §2.1.14.3. For the two-phase problem of interest here, the

governing equations reduce to

$$\begin{aligned}\frac{du_a}{dt} &= -\frac{c}{\tau}(u_a - u_b) \\ \frac{du_b}{dt} &= \frac{1}{\tau}(u_a - u_b)\end{aligned}$$

where $c = \frac{x_b}{x_a}$. Since the mass fractions do not change during the simulation, c is a constant in this problem.

In §3.2.6 the relaxation time, τ , was constant. In the current problem however, τ is no longer a constant, but is determined by the definition

$$\tau = \frac{4 \rho_b}{3 \rho_a C_D} \frac{d_b}{(u_a - u_b)}$$

where d_b is the diameter of the particle. The drag coefficient (C_D) depends on flow properties of the fluid surrounding the particle, such as Reynolds number, Mach number, and porosity. As such, τ varies with time and the analytic expression used for comparison in §3.2.6 is not valid. Therefore, rather than obtaining an analytic expression for comparison, an accurate numerical solution of the above system of ODE's is used. The numerical solution is calculated using the high-order Adams method in SciPy [91]. Since the method used to integrate the multiphase drag terms is expected to be at most first-order accurate in time, the numerical solution is an adequate representation of the *exact* theoretical solution.

Test Results

The calculated velocities show good agreement with the high-order numerical solution. Calculations were performed using the backward-Euler and the analytical integration methods. The simulations were all run over a time interval of one using constant time-steps. In order to show the temporal convergence of the schemes, the simulations used three different time-step sizes, such that 10, 20, and 40 time-steps were used for the simulations. The tests were repeated for three different porosity levels to see the effect of volume fraction.

The results for each drag coefficient law are plotted in the following figures. Each figure contains six subplots which are organized in a table with two columns

and three rows. Each subplot compares the high-order solution (solid line) to the current algorithm's solution (colored symbols) for three different time-step sizes. The left column contains the results using the backward-Euler time integration, while the right column uses the analytic time integration. Each row plots the data from a specific porosity level: the top row contains data for a gas mass fraction of 95%, the middle row 70%, and the bottom row 20%. The drag models tested are the fixed relaxation constant (Figure 3.80), constant drag coefficient (Figure 3.81), Stokes (Figure 3.82), Schiller (Figure 3.83), Putnam (Figure 3.84), Ergun (Figure 3.85), RUC (Figure 3.86), Gidaspow (Figure 3.87), and Akhatov (Figure 3.88).

As would be expected, the results for a fixed relaxation constant are similar to those obtained in §3.2.6. The backward-Euler method converges to the theoretical solution as the time-step size is reduced. The analytic integration method reproduced the exact solution for all time-refinement levels.

Good agreement is also obtained using a constant drag coefficient. Since the relaxation time is now time dependent, the analytical solution method no longer reproduces the high-order solution. However, the analytical method does converge toward the exact solution with time-step refinement. It is noticed that the backward-Euler method has very little error for this case.

Notice that the Stokes drag law is a special case of a fixed relaxation constant. That is because the Stokes drag coefficient is inversely proportional to Reynolds number and hence velocity. Thus the velocity in the definition of τ cancels out, resulting in τ being constant. Therefore, the analytic integration reproduces the exact solution for Stokes drag.

The Schiller and Putnam models are Reynolds-number dependent. Both integration methods exhibit convergence, but it appears that the analytic integration over-predicts the change in velocity, while the backward Euler does a better job. This is especially evident for cases with small relaxation times (faster velocity response), for example Figure 3.84(f). This is because the solution is highly

non-linear in velocity and the assumption that τ is constant over the time-step is less valid. Apparently the implicit formulation of the backward-Euler method handles the non-linearity better.

The Ergun, RUC, Gidaspow, and Akhatov models depend on the porosity of the mixture. As with the other models, these solutions appear adequate and seem to converge to the high-order solution with time refinement.

Test Error Quantification and Convergence

The discretization error for the solution can be quantified using the high-accuracy numerical solution. Since there is no spatial variation in the solution to this problem, spatial-discretization error does not contribute to the total error of the calculation and the time-discretization error should dominate.

The error (E_N) is determined for time-step refinement level N using the methods of §E. The calculated error has been computed for the velocity of both phases separately. These errors are tabulated in Tables 3.19–3.27. The convergence rates (r_N) are also tabulated. These tables are organized in rows and columns similar to the plots discussed earlier in this section.

It is expected that the methods should produce at least first-order convergence in each of the tests. In some cases (for example the analytic integration method for fixed relaxation constants) the results are essentially exact, and therefore the errors are dominated by round-off and the solutions do not exhibit convergence. In other cases the solutions appear to be converging, but the solutions do not appear to have reached the asymptotic regime and so the convergence is less than first order. It is expected that running additional cases with smaller time-step would exhibit first-order behavior. This was verified by running the tests with 1000, 2000, and 4000 time-step where first-order convergence was observed in all cases.

Test Conclusions

The following conclusions can be drawn from this calculation:

- The numerical results seem to adequately predict the theoretical solutions for all drag laws. In general, the solutions appear to exhibit first-order

convergence behavior as expected.

- For non-linear drag laws, the backward-Euler integration method seems to produce better results than the analytic integration method.

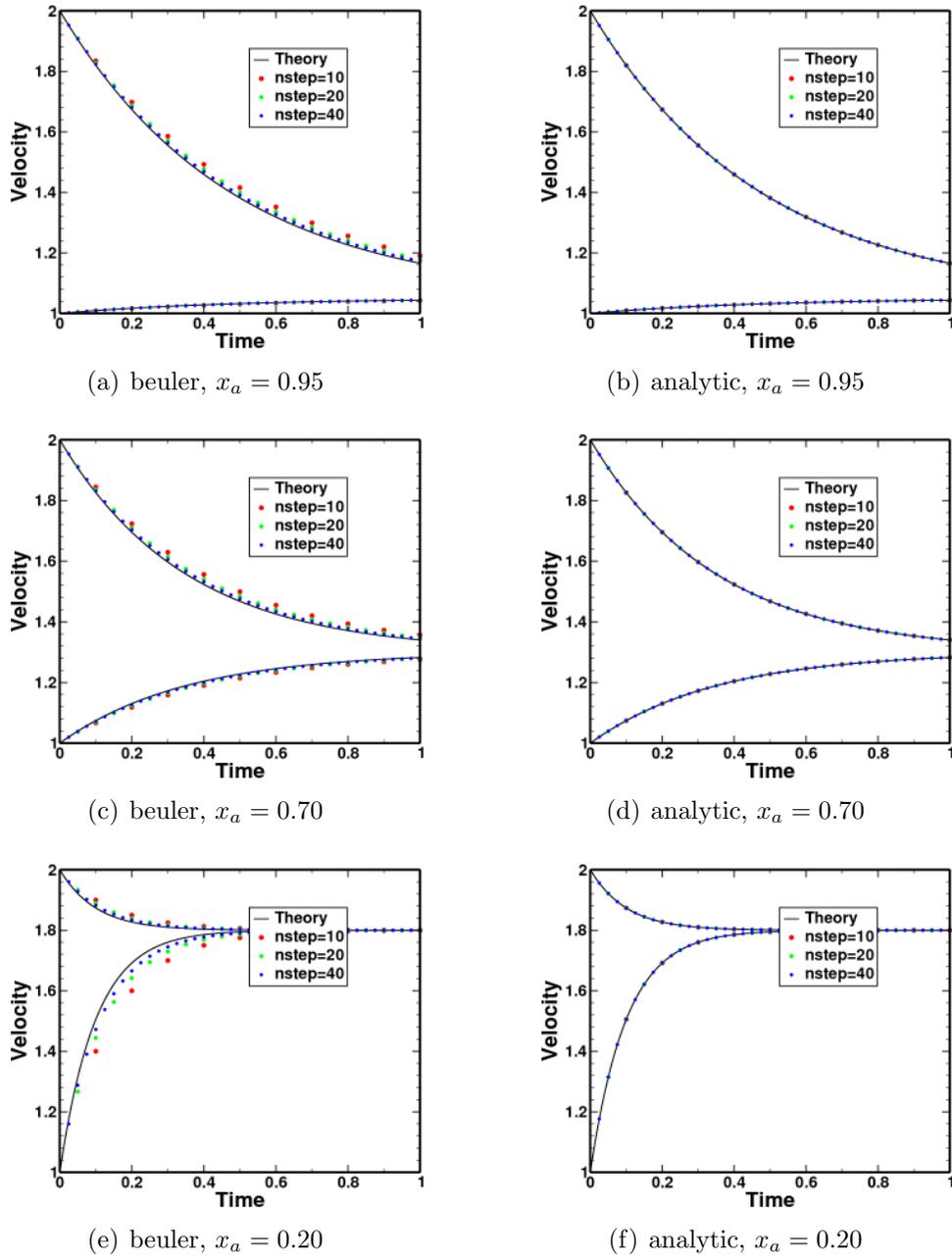


Figure 3.80: Multiphase numerical results for the drag-coefficient tests. The drag-coefficient law used a *fixed relaxation constant*. The higher velocity curves represent the solid (phase *b*) and the lower velocity curves represent the gas (phase *a*).

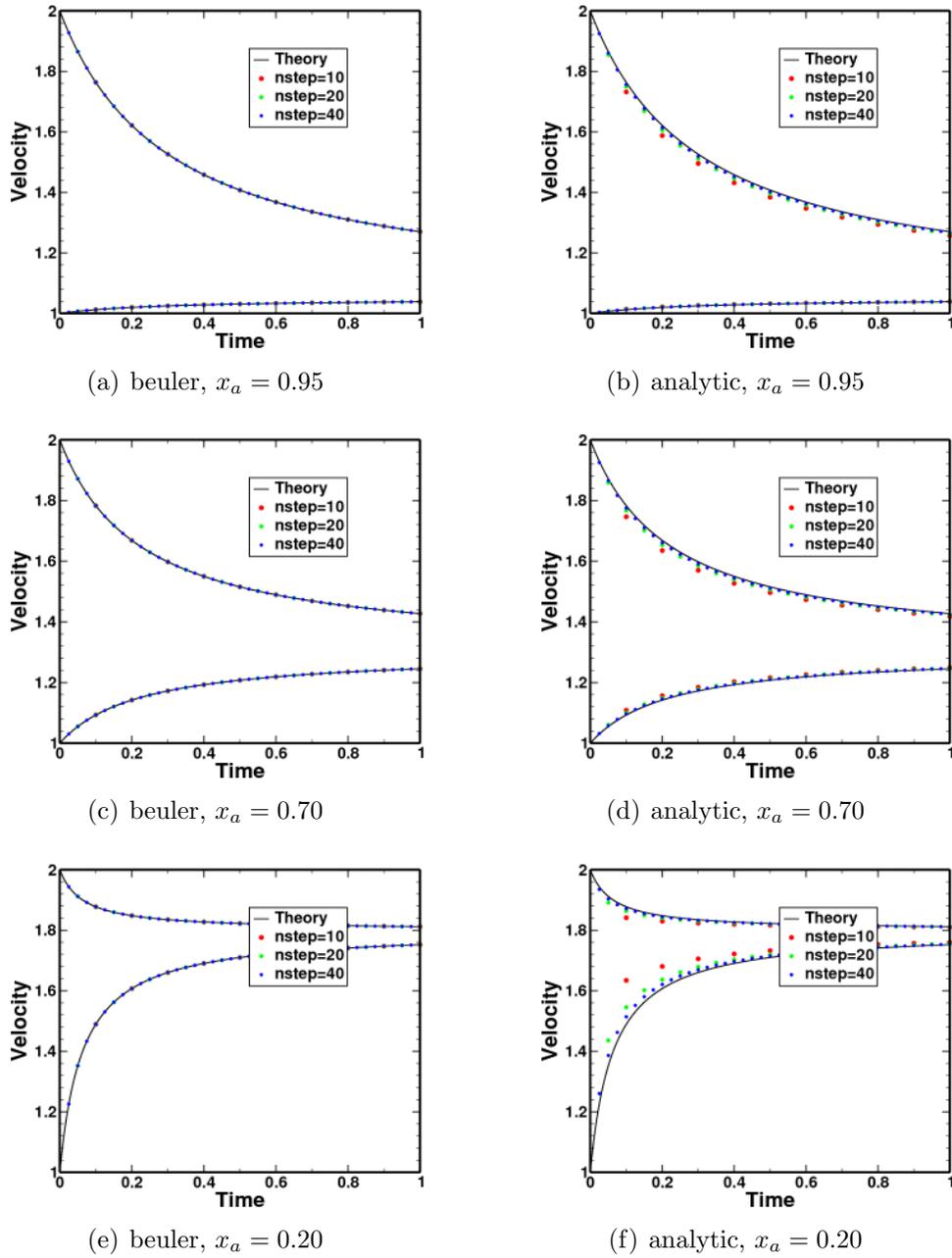


Figure 3.81: Multiphase numerical results for the drag-coefficient tests. The drag-coefficient law used a *constant drag coefficient*. The higher velocity curves represent the solid (phase b) and the lower velocity curves represent the gas (phase a).

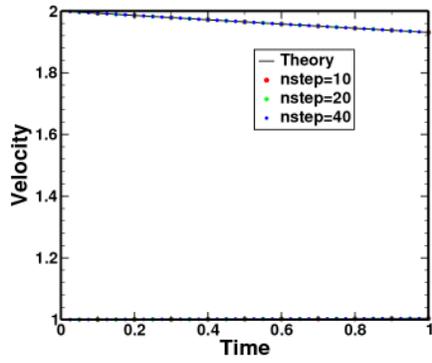
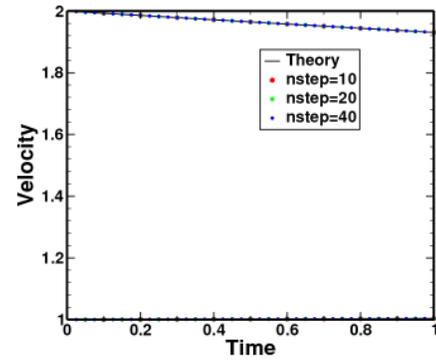
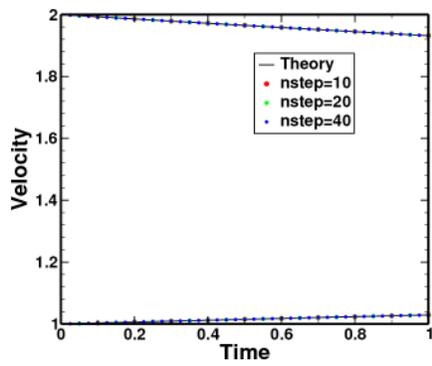
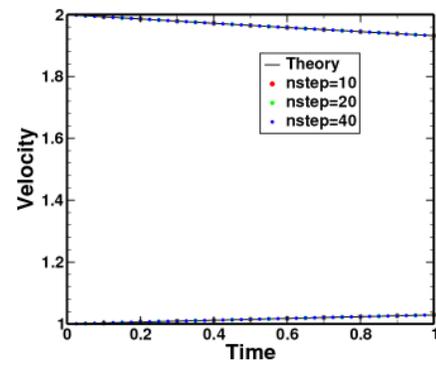
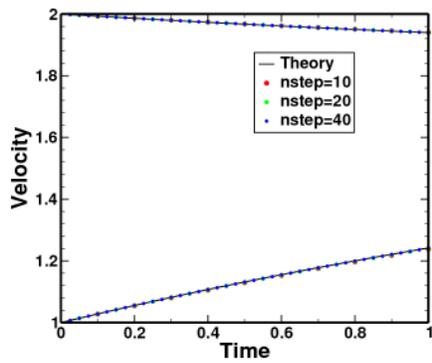
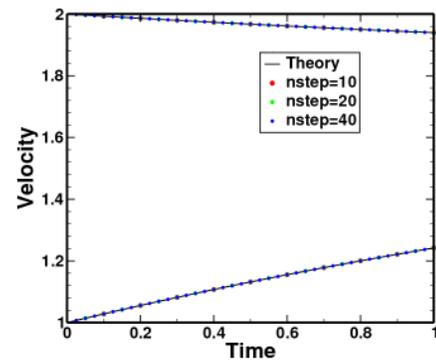
(a) beuler, $x_a = 0.95$ (b) analytic, $x_a = 0.95$ (c) beuler, $x_a = 0.70$ (d) analytic, $x_a = 0.70$ (e) beuler, $x_a = 0.20$ (f) analytic, $x_a = 0.20$

Figure 3.82: Multiphase numerical results for the drag-coefficient tests. The drag-coefficient law used a *Stokes correlation*. The higher velocity curves represent the solid (phase b) and the lower velocity curves represent the gas (phase a).

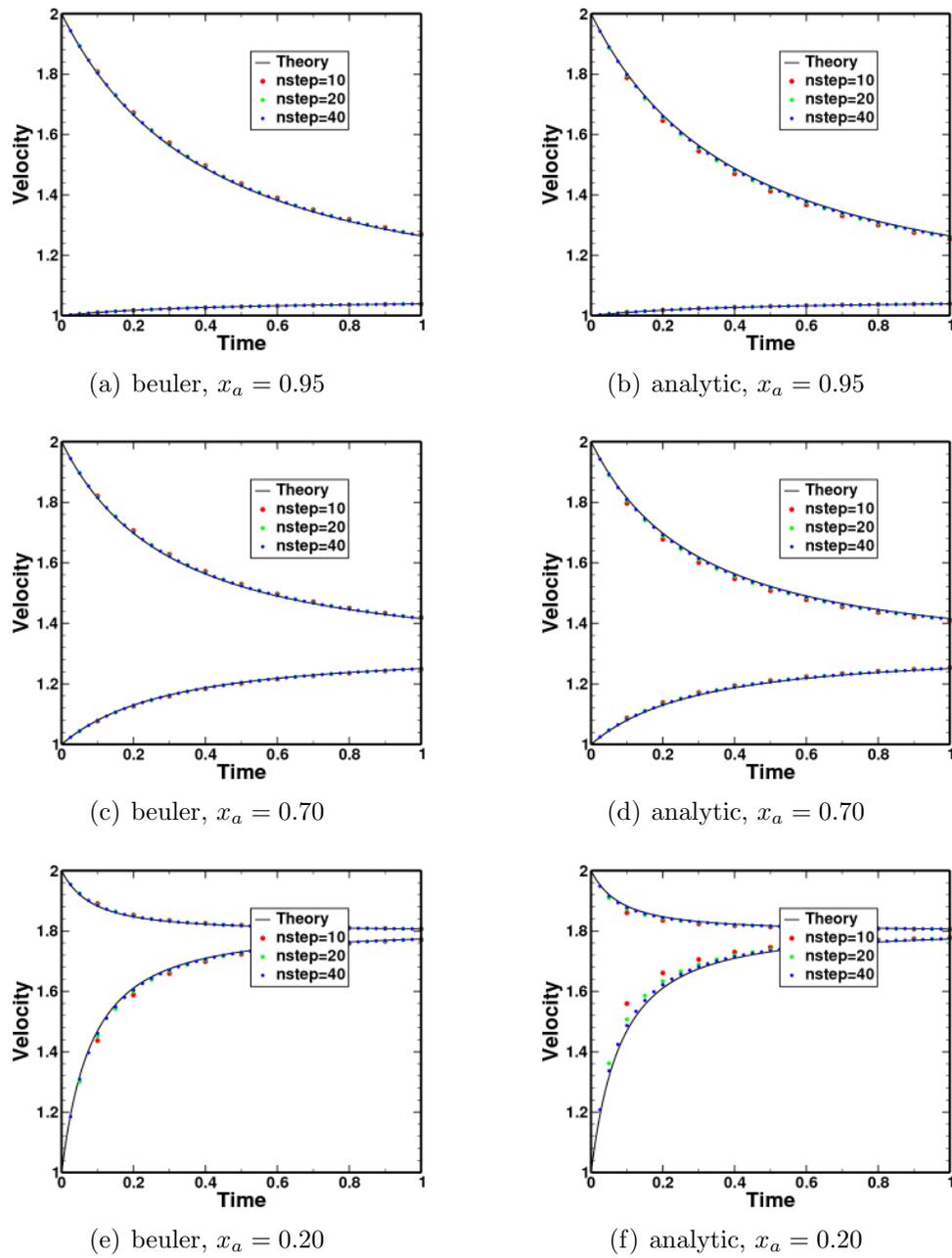


Figure 3.83: Multiphase numerical results for the drag-coefficient tests. The drag-coefficient law used a *Schiller correlation*. The higher velocity curves represent the solid (phase *b*) and the lower velocity curves represent the gas (phase *a*).

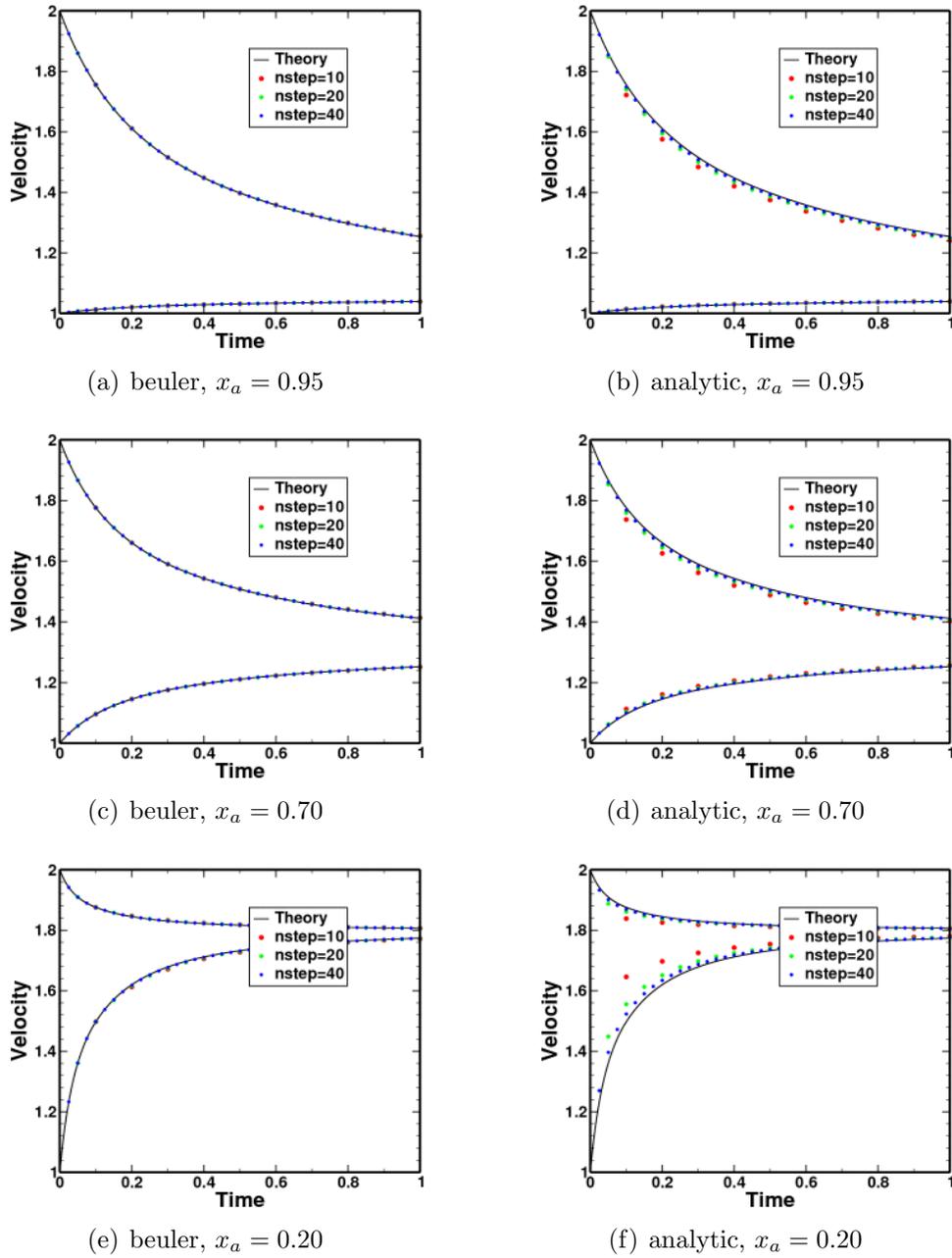


Figure 3.84: Multiphase numerical results for the drag-coefficient tests. The drag-coefficient law used a *Putnam correlation*. The higher velocity curves represent the solid (phase *b*) and the lower velocity curves represent the gas (phase *a*).

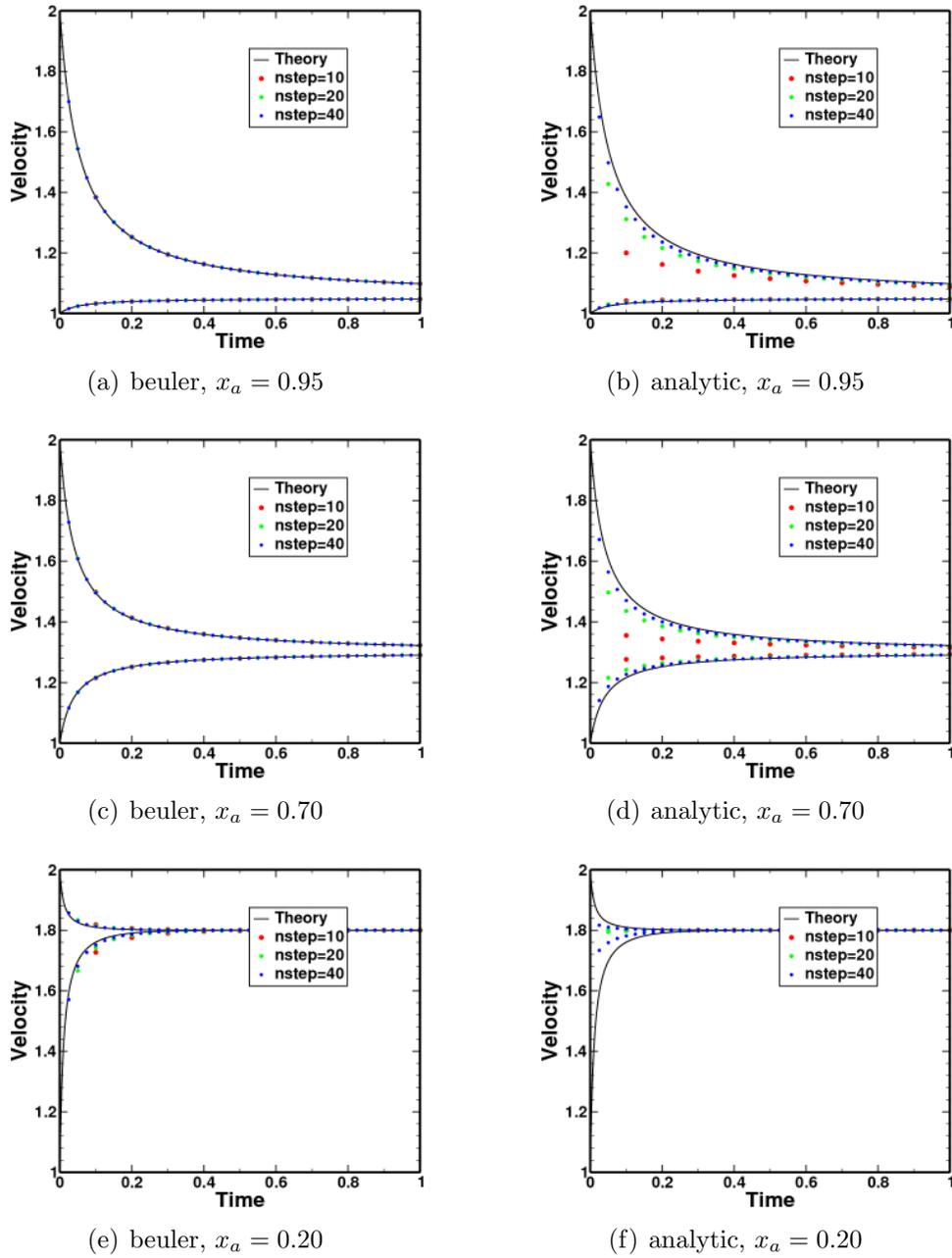


Figure 3.85: Multiphase numerical results for the drag-coefficient tests. The drag-coefficient law used a *Ergun correlation*. The higher velocity curves represent the solid (phase *b*) and the lower velocity curves represent the gas (phase *a*).

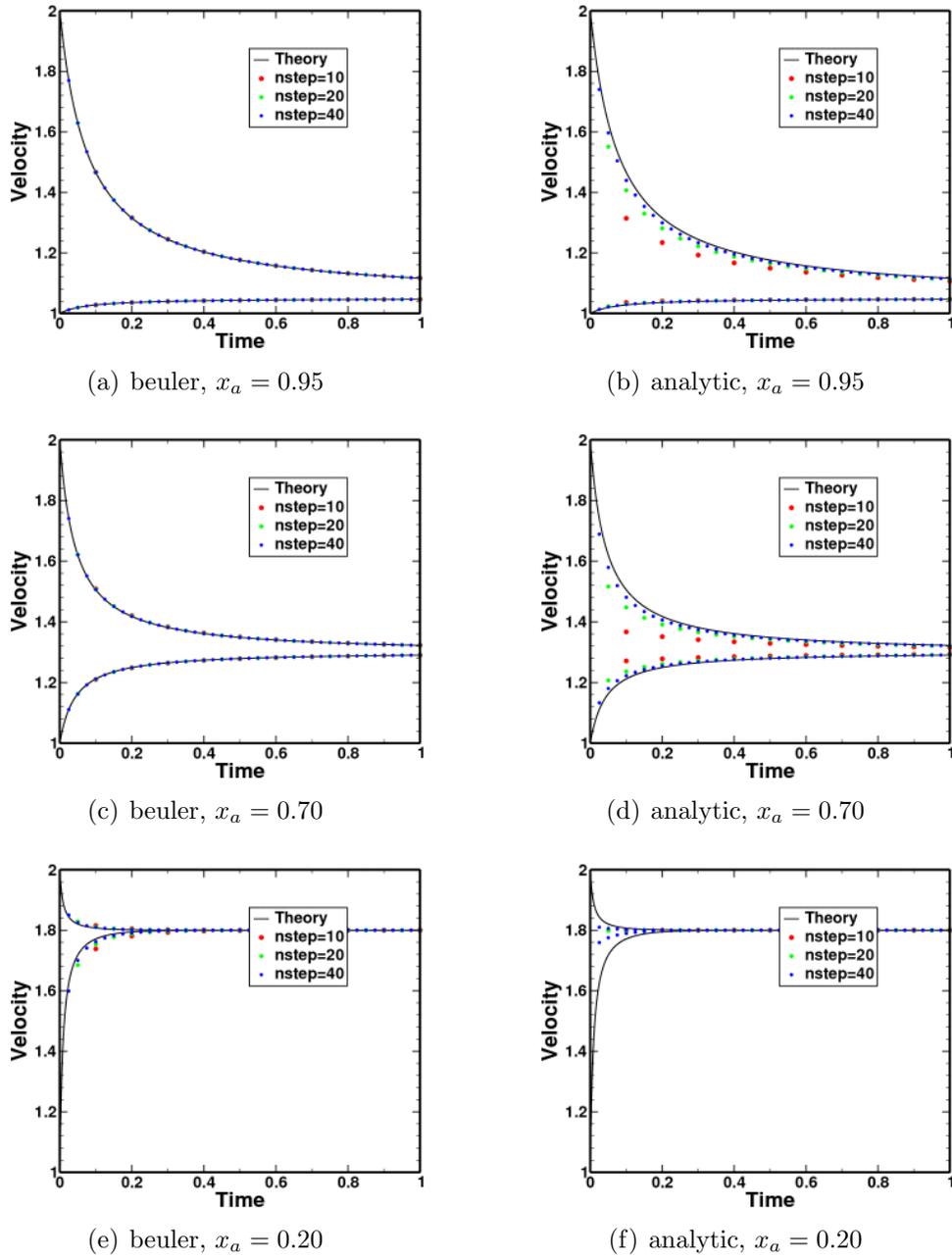


Figure 3.86: Multiphase numerical results for the drag-coefficient tests. The drag-coefficient law used a *RUC correlation*. The higher velocity curves represent the solid (phase *b*) and the lower velocity curves represent the gas (phase *a*).

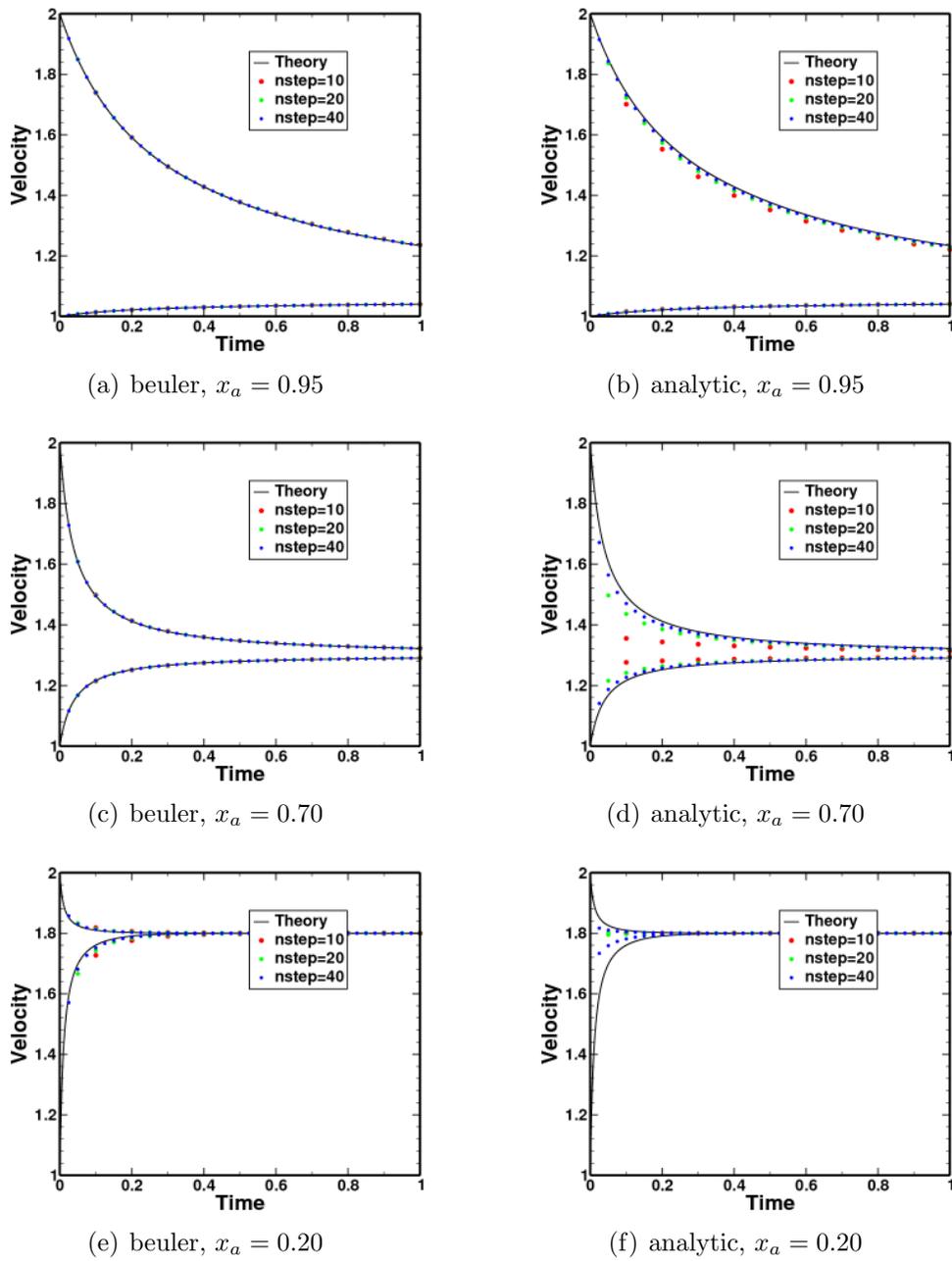


Figure 3.87: Multiphase numerical results for the drag-coefficient tests. The drag-coefficient law used a *Gidaspow correlation*. The higher velocity curves represent the solid (phase *b*) and the lower velocity curves represent the gas (phase *a*).

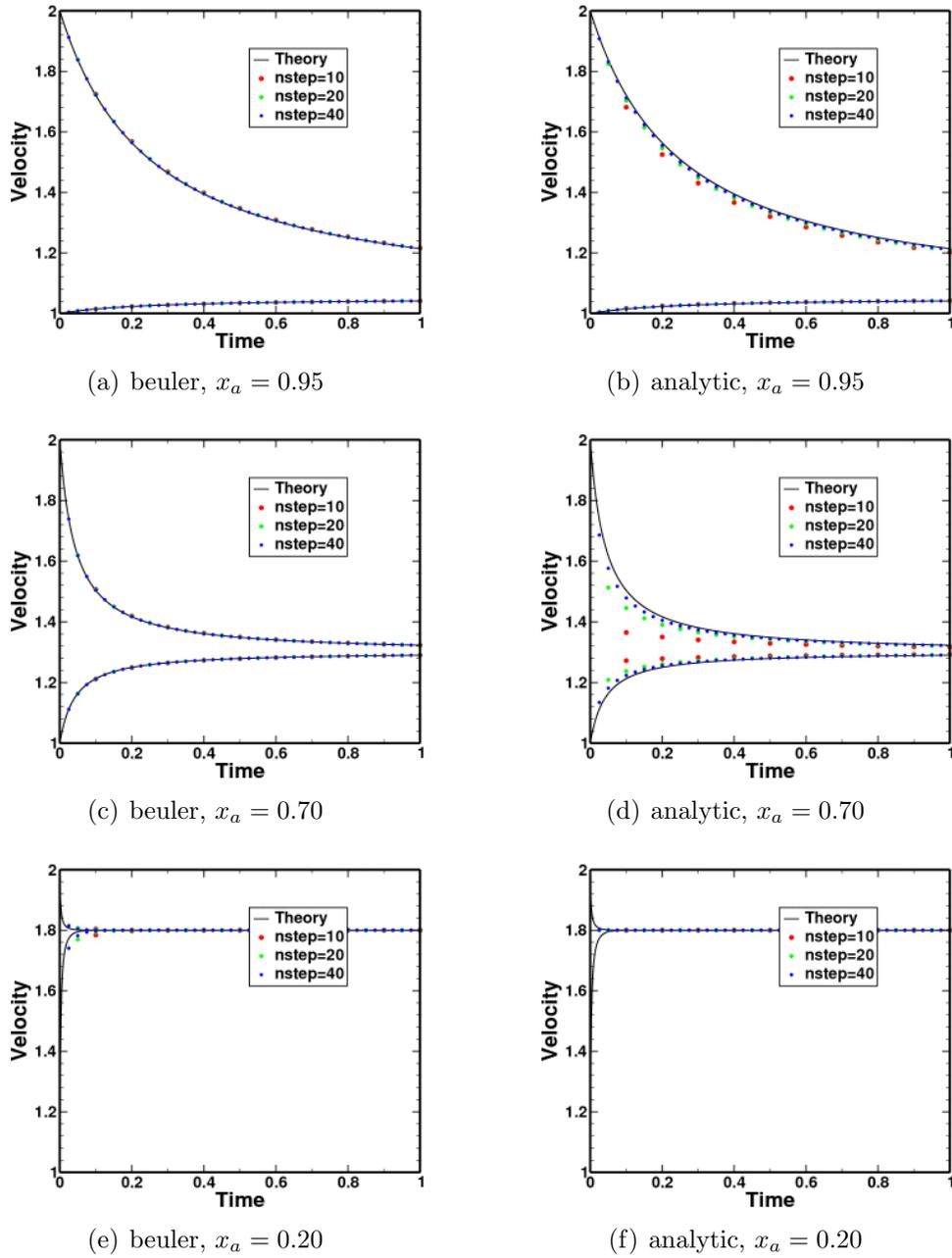


Figure 3.88: Multiphase numerical results for the drag-coefficient tests. The drag-coefficient law used a *Akhatov correlation*. The higher velocity curves represent the solid (phase *b*) and the lower velocity curves represent the gas (phase *a*).

N	E_N	r_N
Phase a Velocity		
10	1.52e-03	
20	7.83e-04	0.96
40	3.97e-04	0.98
Phase b Velocity		
10	2.90e-02	
20	1.49e-02	0.96
40	7.55e-03	0.98

(a) beuler, $x_a = 0.95$

N	E_N	r_N
Phase a Velocity		
10	1.36e-09	
20	1.13e-09	0.27
40	1.12e-09	0.0043
Phase b Velocity		
10	2.59e-08	
20	2.14e-08	0.27
40	2.13e-08	0.0043

(b) analytic, $x_a = 0.95$

N	E_N	r_N
Phase a Velocity		
10	1.13e-02	
20	5.85e-03	0.95
40	2.98e-03	0.97
Phase b Velocity		
10	2.63e-02	
20	1.36e-02	0.95
40	6.96e-03	0.97

(c) beuler, $x_a = 0.70$

N	E_N	r_N
Phase a Velocity		
10	1.11e-08	
20	9.66e-09	0.2
40	9.54e-09	0.019
Phase b Velocity		
10	2.59e-08	
20	2.25e-08	0.2
40	2.23e-08	0.019

(d) analytic, $x_a = 0.70$

N	E_N	r_N
Phase a Velocity		
10	5.00e-02	
20	2.79e-02	0.84
40	1.48e-02	0.91
Phase b Velocity		
10	1.25e-02	
20	6.97e-03	0.84
40	3.70e-03	0.91

(e) beuler, $x_a = 0.20$

N	E_N	r_N
Phase a Velocity		
10	9.81e-09	
20	9.59e-09	0.033
40	1.02e-08	-0.096
Phase b Velocity		
10	2.45e-09	
20	2.40e-09	0.033
40	2.56e-09	-0.096

(f) analytic, $x_a = 0.20$

Table 3.19: Numerical error and convergence rates for the drag-coefficient tests. The drag-coefficient law used a *fixed relaxation constant*.

N	E_N	r_N
Phase a Velocity		
10	1.75e-09	
20	1.75e-09	-0.0041
40	1.73e-09	0.014
Phase b Velocity		
10	3.32e-08	
20	3.33e-08	-0.0041
40	3.30e-08	0.014

(a) beuler, $x_a = 0.95$

N	E_N	r_N
Phase a Velocity		
10	1.26e-03	
20	5.93e-04	1.1
40	2.88e-04	1
Phase b Velocity		
10	2.40e-02	
20	1.13e-02	1.1
40	5.46e-03	1

(b) analytic, $x_a = 0.95$

N	E_N	r_N
Phase a Velocity		
10	1.47e-08	
20	1.43e-08	0.036
40	1.43e-08	0.0021
Phase b Velocity		
10	3.42e-08	
20	3.34e-08	0.036
40	3.33e-08	0.0021

(c) beuler, $x_a = 0.70$

N	E_N	r_N
Phase a Velocity		
10	9.50e-03	
20	4.39e-03	1.1
40	2.11e-03	1.1
Phase b Velocity		
10	2.22e-02	
20	1.02e-02	1.1
40	4.93e-03	1.1

(d) analytic, $x_a = 0.70$

N	E_N	r_N
Phase a Velocity		
10	3.44e-08	
20	3.42e-08	0.0094
40	3.45e-08	-0.0097
Phase b Velocity		
10	8.61e-09	
20	8.56e-09	0.0094
40	8.61e-09	-0.0097

(e) beuler, $x_a = 0.20$

N	E_N	r_N
Phase a Velocity		
10	5.55e-02	
20	2.69e-02	1
40	1.21e-02	1.1
Phase b Velocity		
10	1.39e-02	
20	6.72e-03	1
40	3.03e-03	1.1

(f) analytic, $x_a = 0.20$

Table 3.20: Numerical error and convergence rates for the drag-coefficient tests. The drag-coefficient law used a *constant drag coefficient*.

N	E_N	r_N
Phase a Velocity		
10	8.36e-06	
20	4.05e-06	1
40	1.99e-06	1
Phase b Velocity		
10	1.59e-04	
20	7.69e-05	1
40	3.78e-05	1

(a) beuler, $x_a = 0.95$

N	E_N	r_N
Phase a Velocity		
10	3.33e-11	
20	2.87e-11	0.22
40	2.64e-11	0.12
Phase b Velocity		
10	6.33e-10	
20	5.46e-10	0.21
40	5.02e-10	0.12

(b) analytic, $x_a = 0.95$

N	E_N	r_N
Phase a Velocity		
10	9.03e-05	
20	4.38e-05	1
40	2.16e-05	1
Phase b Velocity		
10	2.11e-04	
20	1.02e-04	1
40	5.03e-05	1

(c) beuler, $x_a = 0.70$

N	E_N	r_N
Phase a Velocity		
10	7.33e-10	
20	6.33e-10	0.21
40	5.83e-10	0.12
Phase b Velocity		
10	1.71e-09	
20	1.48e-09	0.21
40	1.36e-09	0.12

(d) analytic, $x_a = 0.70$

N	E_N	r_N
Phase a Velocity		
10	2.39e-03	
20	1.17e-03	1
40	5.80e-04	1
Phase b Velocity		
10	5.97e-04	
20	2.93e-04	1
40	1.45e-04	1

(e) beuler, $x_a = 0.20$

N	E_N	r_N
Phase a Velocity		
10	2.18e-09	
20	1.79e-09	0.28
40	1.60e-09	0.16
Phase b Velocity		
10	5.45e-10	
20	4.48e-10	0.28
40	4.00e-10	0.16

(f) analytic, $x_a = 0.20$

Table 3.21: Numerical error and convergence rates for the drag-coefficient tests. The drag-coefficient law used a *Stokes correlation*.

N	E_N	r_N
Phase a Velocity		
10	3.89e-04	
20	1.93e-04	1
40	9.65e-05	1
Phase b Velocity		
10	7.38e-03	
20	3.68e-03	1
40	1.83e-03	1

(a) beuler, $x_a = 0.95$

N	E_N	r_N
Phase a Velocity		
10	8.04e-04	
20	3.87e-04	1.1
40	1.90e-04	1
Phase b Velocity		
10	1.53e-02	
20	7.34e-03	1.1
40	3.60e-03	1

(b) analytic, $x_a = 0.95$

N	E_N	r_N
Phase a Velocity		
10	2.89e-03	
20	1.44e-03	1
40	7.19e-04	1
Phase b Velocity		
10	6.74e-03	
20	3.36e-03	1
40	1.68e-03	1

(c) beuler, $x_a = 0.70$

N	E_N	r_N
Phase a Velocity		
10	6.06e-03	
20	2.89e-03	1.1
40	1.41e-03	1
Phase b Velocity		
10	1.41e-02	
20	6.75e-03	1.1
40	3.30e-03	1

(d) analytic, $x_a = 0.70$

N	E_N	r_N
Phase a Velocity		
10	1.46e-02	
20	7.74e-03	0.92
40	3.90e-03	0.99
Phase b Velocity		
10	3.65e-03	
20	1.93e-03	0.92
40	9.76e-04	0.99

(e) beuler, $x_a = 0.20$

N	E_N	r_N
Phase a Velocity		
10	3.53e-02	
20	1.69e-02	1.1
40	7.93e-03	1.1
Phase b Velocity		
10	8.82e-03	
20	4.22e-03	1.1
40	1.98e-03	1.1

(f) analytic, $x_a = 0.20$

Table 3.22: Numerical error and convergence rates for the drag-coefficient tests. The drag-coefficient law used a *Schiller correlation*.

N	E_N	r_N
Phase a Velocity		
10	4.69e-05	
20	2.82e-05	0.74
40	1.17e-05	1.3
Phase b Velocity		
10	8.91e-04	
20	5.35e-04	0.74
40	2.23e-04	1.3

(a) beuler, $x_a = 0.95$

N	E_N	r_N
Phase a Velocity		
10	1.30e-03	
20	6.17e-04	1.1
40	2.95e-04	1.1
Phase b Velocity		
10	2.47e-02	
20	1.17e-02	1.1
40	5.60e-03	1.1

(b) analytic, $x_a = 0.95$

N	E_N	r_N
Phase a Velocity		
10	4.77e-04	
20	2.56e-04	0.9
40	1.51e-04	0.76
Phase b Velocity		
10	1.11e-03	
20	5.97e-04	0.9
40	3.52e-04	0.76

(c) beuler, $x_a = 0.70$

N	E_N	r_N
Phase a Velocity		
10	9.82e-03	
20	4.49e-03	1.1
40	2.19e-03	1
Phase b Velocity		
10	2.29e-02	
20	1.05e-02	1.1
40	5.11e-03	1

(d) analytic, $x_a = 0.70$

N	E_N	r_N
Phase a Velocity		
10	4.65e-03	
20	2.32e-03	1
40	1.13e-03	1
Phase b Velocity		
10	1.16e-03	
20	5.80e-04	1
40	2.82e-04	1

(e) beuler, $x_a = 0.20$

N	E_N	r_N
Phase a Velocity		
10	5.67e-02	
20	2.78e-02	1
40	1.25e-02	1.1
Phase b Velocity		
10	1.42e-02	
20	6.96e-03	1
40	3.14e-03	1.1

(f) analytic, $x_a = 0.20$

Table 3.23: Numerical error and convergence rates for the drag-coefficient tests. The drag-coefficient law used a *Putnam correlation*.

N	E_N	r_N
Phase a Velocity		
10	9.65e-06	
20	5.10e-06	0.92
40	2.58e-06	0.98
Phase b Velocity		
10	1.83e-04	
20	9.69e-05	0.92
40	4.90e-05	0.98

(a) beuler, $x_a = 0.95$

N	E_N	r_N
Phase a Velocity		
10	3.65e-03	
20	1.86e-03	0.98
40	8.38e-04	1.1
Phase b Velocity		
10	6.94e-02	
20	3.53e-02	0.98
40	1.59e-02	1.1

(b) analytic, $x_a = 0.95$

N	E_N	r_N
Phase a Velocity		
10	5.31e-04	
20	2.91e-04	0.87
40	1.49e-04	0.96
Phase b Velocity		
10	1.24e-03	
20	6.78e-04	0.87
40	3.49e-04	0.96

(c) beuler, $x_a = 0.70$

N	E_N	r_N
Phase a Velocity		
10	2.26e-02	
20	1.32e-02	0.78
40	6.13e-03	1.1
Phase b Velocity		
10	5.27e-02	
20	3.08e-02	0.78
40	1.43e-02	1.1

(d) analytic, $x_a = 0.70$

N	E_N	r_N
Phase a Velocity		
10	1.19e-02	
20	8.79e-03	0.44
40	5.59e-03	0.65
Phase b Velocity		
10	2.97e-03	
20	2.20e-03	0.44
40	1.40e-03	0.65

(e) beuler, $x_a = 0.20$

N	E_N	r_N
Phase a Velocity		
10	1.29e-02	
20	2.36e-02	-0.87
40	2.49e-02	-0.082
Phase b Velocity		
10	3.22e-03	
20	5.89e-03	-0.87
40	6.24e-03	-0.082

(f) analytic, $x_a = 0.20$

Table 3.24: Numerical error and convergence rates for the drag-coefficient tests. The drag-coefficient law used a *Ergun correlation*.

N	E_N	r_N
Phase a Velocity		
10	2.79e-05	
20	1.43e-05	0.96
40	7.20e-06	0.99
Phase b Velocity		
10	5.30e-04	
20	2.72e-04	0.96
40	1.37e-04	0.99

(a) beuler, $x_a = 0.95$

N	E_N	r_N
Phase a Velocity		
10	3.13e-03	
20	1.46e-03	1.1
40	6.63e-04	1.1
Phase b Velocity		
10	5.96e-02	
20	2.77e-02	1.1
40	1.26e-02	1.1

(b) analytic, $x_a = 0.95$

N	E_N	r_N
Phase a Velocity		
10	7.57e-04	
20	4.11e-04	0.88
40	2.11e-04	0.96
Phase b Velocity		
10	1.77e-03	
20	9.60e-04	0.88
40	4.92e-04	0.96

(c) beuler, $x_a = 0.70$

N	E_N	r_N
Phase a Velocity		
10	2.23e-02	
20	1.26e-02	0.82
40	5.79e-03	1.1
Phase b Velocity		
10	5.19e-02	
20	2.93e-02	0.82
40	1.35e-02	1.1

(d) analytic, $x_a = 0.70$

N	E_N	r_N
Phase a Velocity		
10	1.15e-02	
20	8.90e-03	0.37
40	5.88e-03	0.6
Phase b Velocity		
10	2.88e-03	
20	2.22e-03	0.37
40	1.47e-03	0.6

(e) beuler, $x_a = 0.20$

N	E_N	r_N
Phase a Velocity		
10	9.25e-03	
20	1.91e-02	-1
40	2.38e-02	-0.31
Phase b Velocity		
10	2.31e-03	
20	4.78e-03	-1
40	5.94e-03	-0.31

(f) analytic, $x_a = 0.20$

Table 3.25: Numerical error and convergence rates for the drag-coefficient tests. The drag-coefficient law used a *RUC correlation*.

N	E_N	r_N
Phase a Velocity		
10	8.56e-05	
20	4.32e-05	0.99
40	2.19e-05	0.98
Phase b Velocity		
10	1.63e-03	
20	8.20e-04	0.99
40	4.17e-04	0.98

(a) beuler, $x_a = 0.95$

N	E_N	r_N
Phase a Velocity		
10	1.39e-03	
20	6.48e-04	1.1
40	3.14e-04	1
Phase b Velocity		
10	2.64e-02	
20	1.23e-02	1.1
40	5.96e-03	1

(b) analytic, $x_a = 0.95$

N	E_N	r_N
Phase a Velocity		
10	5.31e-04	
20	2.91e-04	0.87
40	1.49e-04	0.96
Phase b Velocity		
10	1.24e-03	
20	6.78e-04	0.87
40	3.49e-04	0.96

(c) beuler, $x_a = 0.70$

N	E_N	r_N
Phase a Velocity		
10	2.26e-02	
20	1.32e-02	0.78
40	6.13e-03	1.1
Phase b Velocity		
10	5.27e-02	
20	3.08e-02	0.78
40	1.43e-02	1.1

(d) analytic, $x_a = 0.70$

N	E_N	r_N
Phase a Velocity		
10	1.19e-02	
20	8.79e-03	0.44
40	5.59e-03	0.65
Phase b Velocity		
10	2.97e-03	
20	2.20e-03	0.44
40	1.40e-03	0.65

(e) beuler, $x_a = 0.20$

N	E_N	r_N
Phase a Velocity		
10	1.29e-02	
20	2.36e-02	-0.87
40	2.49e-02	-0.082
Phase b Velocity		
10	3.22e-03	
20	5.89e-03	-0.87
40	6.24e-03	-0.082

(f) analytic, $x_a = 0.20$

Table 3.26: Numerical error and convergence rates for the drag-coefficient tests. The drag-coefficient law used a *Gidaspow correlation*.

N	E_N	r_N
Phase a Velocity		
10	1.83e-04	
20	9.10e-05	1
40	4.53e-05	1
Phase b Velocity		
10	3.47e-03	
20	1.73e-03	1
40	8.61e-04	1

(a) beuler, $x_a = 0.95$

N	E_N	r_N
Phase a Velocity		
10	1.37e-03	
20	6.41e-04	1.1
40	3.10e-04	1
Phase b Velocity		
10	2.60e-02	
20	1.22e-02	1.1
40	5.89e-03	1

(b) analytic, $x_a = 0.95$

N	E_N	r_N
Phase a Velocity		
10	7.13e-04	
20	3.92e-04	0.86
40	2.01e-04	0.96
Phase b Velocity		
10	1.66e-03	
20	9.15e-04	0.86
40	4.69e-04	0.96

(c) beuler, $x_a = 0.70$

N	E_N	r_N
Phase a Velocity		
10	2.23e-02	
20	1.27e-02	0.82
40	5.84e-03	1.1
Phase b Velocity		
10	5.20e-02	
20	2.95e-02	0.82
40	1.36e-02	1.1

(d) analytic, $x_a = 0.70$

N	E_N	r_N
Phase a Velocity		
10	4.95e-03	
20	5.96e-03	-0.27
40	5.65e-03	0.078
Phase b Velocity		
10	1.24e-03	
20	1.49e-03	-0.27
40	1.41e-03	0.078

(e) beuler, $x_a = 0.20$

N	E_N	r_N
Phase a Velocity		
10	7.58e-05	
20	1.12e-03	-3.9
40	4.31e-03	-1.9
Phase b Velocity		
10	1.89e-05	
20	2.81e-04	-3.9
40	1.08e-03	-1.9

(f) analytic, $x_a = 0.20$

Table 3.27: Numerical error and convergence rates for the drag-coefficient tests. The drag-coefficient law used a *Akhatov correlation*.

3.2.8 Multiphase Drag with Conduction

Test Description

This tests the multiphase drag and heat-transfer models. It is a single-element problem with a gas and solid phase. The only interaction between the phases occurs due to temperature and velocity relaxation.

Test Setup

This problem consists of a single gas phase (*a*) and a single solid phase (*b*). The geometry is a single element with outflow boundary conditions enforced on all external faces. Thus, there are no flow gradients possible and the hydrodynamic equations are trivially satisfied. Initially, the phases are at different velocities, resulting in momentum exchange through drag. The gas phase is initially at velocity $\overline{\overline{U}}_a = 1$ and the solid phase is initially at velocity $\overline{\overline{U}}_b = 2$. There is also a temperature difference between the phases, resulting in conductive heat transfer. The gas phase is initially at temperature $\overline{\overline{T}}_a = 2$ and the solid phase is initially at temperature $\overline{\overline{T}}_b = 3$.

To complicate the problem further, temperature-dependent material properties are used. The following relations are used for the specific heat, thermal conductivity, and viscosity:

$$C_{va} = 2.5 + 2.5T_a$$

$$C_{vb} = 0.4 + 0.4T_b$$

$$k_a = 0.0001 + 0.0001T_a$$

$$\mu_a = 0.0001T_a^2$$

The problem is run with the Ranz-Marshall relation for the Nusselt number (see §C) and the Ergun equation for the drag coefficient (see §B). Therefore, both the drag and heat transfer are functions of Reynolds number. The tests are repeated for three different porosity levels with gas mass fractions of 95%, 70%, and 20%.

Test Theory

The analytical expression for the velocity and temperature evolution of a multiphase mixture from drag and heat transfer between the phases is discussed in §2.1.14.2 and §2.1.14.3. For the two-phase problem of interest here, the governing equations reduce to

$$\begin{aligned}\frac{du_a}{dt} &= -\frac{x_b}{x_a\tau_v}(u_a - u_b) \\ \frac{du_b}{dt} &= \frac{1}{\tau_v}(u_a - u_b) \\ \frac{dT_a}{dt} &= -\frac{x_b C_{vb}}{x_a C_{va}\tau_t}(T_a - T_b) + \frac{u_a}{C_{va}}\frac{du_a}{dt} + \frac{x_b u_b}{x_a C_{va}}\frac{du_b}{dt} \\ \frac{dT_b}{dt} &= \frac{1}{\tau_t}(T_a - T_b)\end{aligned}$$

where the mass fractions, x_a and x_b , are constants in the current problem. The extra terms on the third equation account for kinetic energy transfer due to drag. The relaxation response times are given by

$$\tau_v = \frac{4\rho_b}{3\rho_a} \frac{d_b}{C_D(u_a - u_b)}$$

for the velocity and

$$\tau_t = \frac{1}{12} \frac{\rho_b C_{vb} d_b^2}{k_a} \frac{2}{Nu}$$

for the temperature.

As there is not an analytic solution for the above coupled system, an accurate numerical solution of the above system of ODE's is used for comparison. The numerical solution is calculated using the high-order Adams method in SciPy [91]. Since the methods used to integrate the multiphase drag and heat-transfer terms are expected to be at most first-order accurate in time, the numerical solution is an adequate representation of the *exact* theoretical solution.

Test Results

The calculated velocity and temperature show good agreement with the high-order numerical solution. Calculations were performed using the backward-Euler

and the analytical integration methods. The simulations were all run over a time interval of one using constant time-steps. In order to show the temporal convergence of the schemes, the simulations used three different time-step sizes, such that 10, 20, and 40 time-steps were used for the simulations. The tests were repeated for three different porosity levels to see the effect of volume fraction.

The results are plotted in the following figures. Each figure contains six subplots which are organized in a table with two columns and three rows. Each subplot compares the high-order solution (solid line) to the current algorithm's solution (colored symbols) for three different time-step sizes. The left column contains the results using the backward-Euler time integration, while the right column uses the analytic time integration. Each row plots the data from a specific porosity level: the top row contains data for a gas mass fraction of 95%, the middle row 70%, and the bottom row 20%. The velocity results are in Figure 3.89 and the temperature results are in Figure 3.90. The model seems to accurately reproduce the high-order solution in all the plots.

Test Error Quantification and Convergence

The discretization error for the solution can be quantified using the high-accuracy numerical solution. Since there is no spatial variation in the solution to this problem, spatial-discretization error does not contribute to the total error of the calculation and the time-discretization error should dominate.

The error (E_N) is determined for time-step refinement level N using the methods of §E. The calculated error has been computed for the temperature and velocity of both phases separately. These errors are tabulated in Tables 3.28–3.29. The convergence rates (r_N) are also tabulated. These tables are organized in rows and columns similar to the plots discussed earlier in this section.

It is expected that the methods should produce at least first-order convergence in each of the tests. First-order convergence is seen in both the analytic and backward-Euler integration methods. In some cases the solutions do not appear to have reached the asymptotic regime and so the convergence is slightly less than

first order. It is expected that running additional cases with smaller time-step would exhibit first-order behavior. This was verified by running the tests with 1000, 2000, and 4000 time-step where first-order convergence was observed in all cases.

Test Conclusions

The following conclusions can be drawn from this calculation:

- The numerical results seem to adequately predict the theoretical solutions for coupled drag-conduction problems. In general, the solutions appear to exhibit first-order convergence behavior as expected.
- The behavior of temperature-dependent material properties appears to be handled correctly.

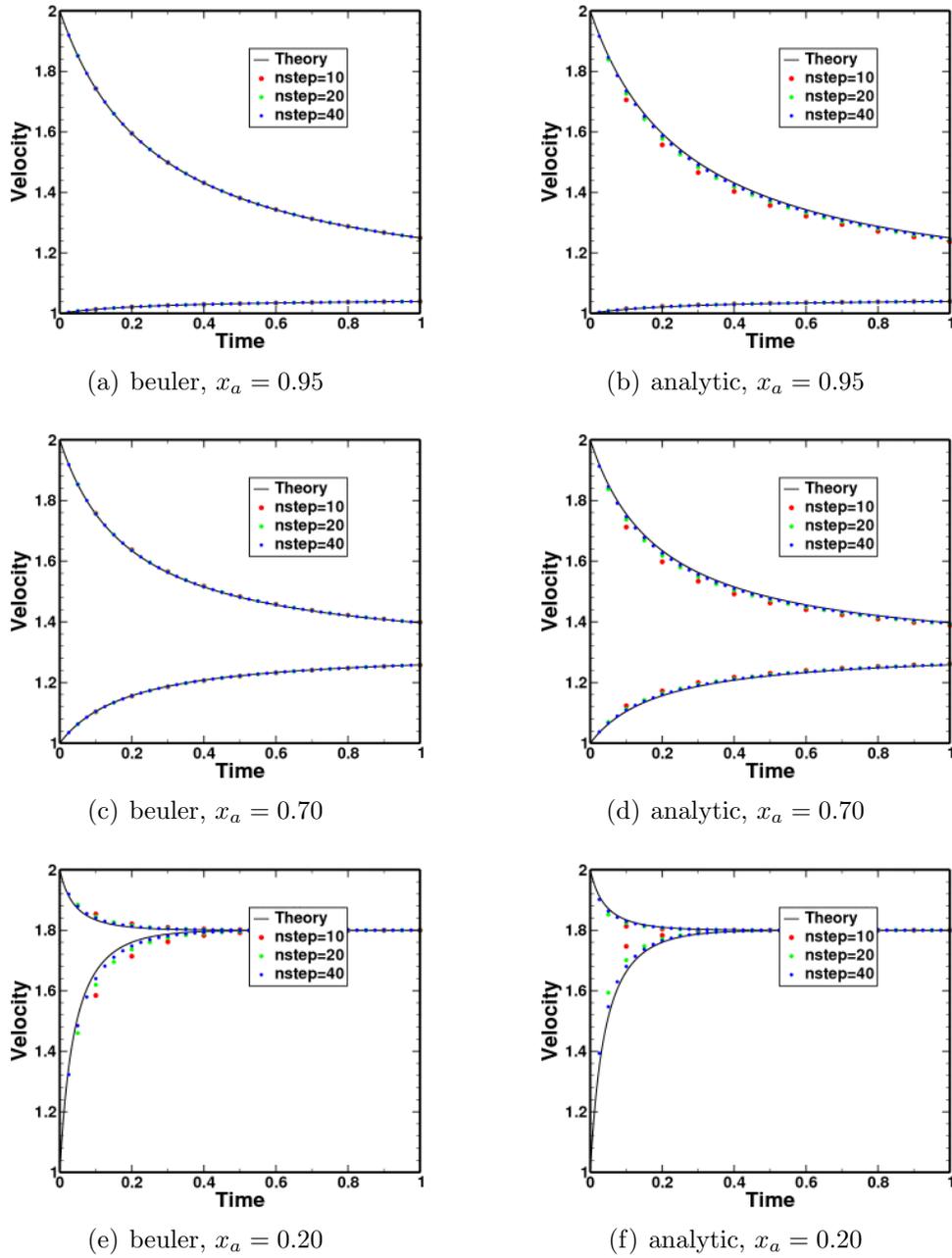


Figure 3.89: Multiphase numerical results for the coupled drag/conduction tests: velocity results. The higher velocity curves represent the solid (phase b) and the lower velocity curves represent the gas (phase a).

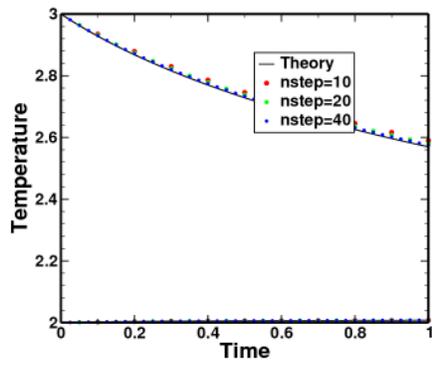
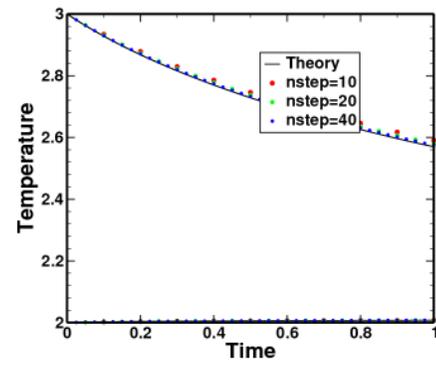
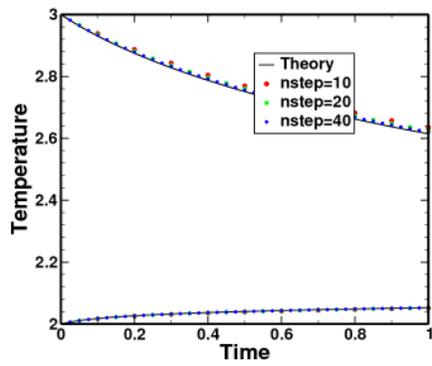
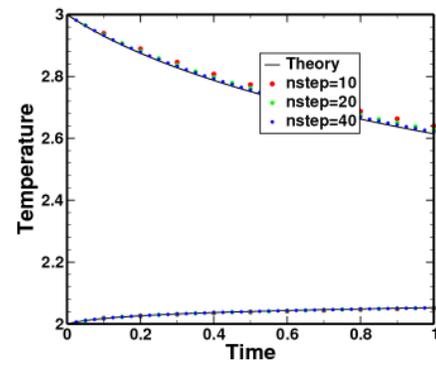
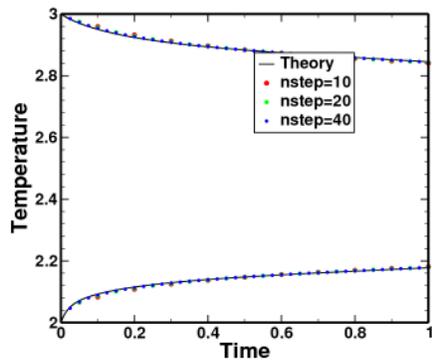
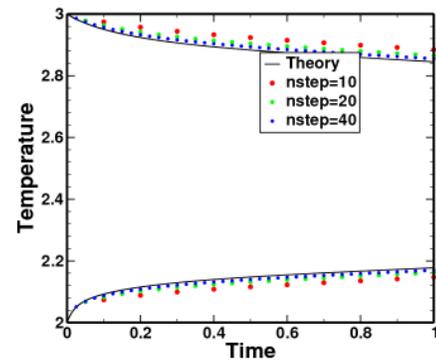
(a) beuler, $x_a = 0.95$ (b) analytic, $x_a = 0.95$ (c) beuler, $x_a = 0.70$ (d) analytic, $x_a = 0.70$ (e) beuler, $x_a = 0.20$ (f) analytic, $x_a = 0.20$

Figure 3.90: Multiphase numerical results for the coupled drag/conduction tests: temperature results. The higher temperature curves represent the solid (phase b) and the lower temperature curves represent the gas (phase a).

N	E_N	r_N
Phase a Velocity		
10	1.43e-05	
20	7.12e-06	1
40	3.55e-06	1
Phase b Velocity		
10	2.72e-04	
20	1.35e-04	1
40	6.74e-05	1

(a) beuler, $x_a = 0.95$

N	E_N	r_N
Phase a Velocity		
10	1.37e-03	
20	6.38e-04	1.1
40	3.09e-04	1
Phase b Velocity		
10	2.60e-02	
20	1.21e-02	1.1
40	5.87e-03	1

(b) analytic, $x_a = 0.95$

N	E_N	r_N
Phase a Velocity		
10	9.34e-04	
20	4.69e-04	0.99
40	2.35e-04	1
Phase b Velocity		
10	2.18e-03	
20	1.09e-03	0.99
40	5.47e-04	1

(c) beuler, $x_a = 0.70$

N	E_N	r_N
Phase a Velocity		
10	1.02e-02	
20	4.68e-03	1.1
40	2.24e-03	1.1
Phase b Velocity		
10	2.37e-02	
20	1.09e-02	1.1
40	5.23e-03	1.1

(d) analytic, $x_a = 0.70$

N	E_N	r_N
Phase a Velocity		
10	3.03e-02	
20	1.87e-02	0.7
40	1.02e-02	0.87
Phase b Velocity		
10	7.59e-03	
20	4.67e-03	0.7
40	2.55e-03	0.87

(e) beuler, $x_a = 0.20$

N	E_N	r_N
Phase a Velocity		
10	2.75e-02	
20	2.05e-02	0.42
40	1.03e-02	0.99
Phase b Velocity		
10	6.88e-03	
20	5.13e-03	0.42
40	2.58e-03	0.99

(f) analytic, $x_a = 0.20$

Table 3.28: Numerical error and convergence rates for the coupled drag/conduction tests: velocity results.

N	E_N	r_N
Phase a Temperature		
10	1.75e-04	
20	8.93e-05	0.97
40	4.52e-05	0.98
Phase b Temperature		
10	1.69e-02	
20	8.61e-03	0.97
40	4.35e-03	0.98

(a) beuler, $x_a = 0.95$

N	E_N	r_N
Phase a Temperature		
10	1.40e-04	
20	6.62e-05	1.1
40	3.22e-05	1
Phase b Temperature		
10	1.70e-02	
20	8.15e-03	1.1
40	3.99e-03	1

(b) analytic, $x_a = 0.95$

N	E_N	r_N
Phase a Temperature		
10	1.48e-03	
20	7.67e-04	0.95
40	3.91e-04	0.97
Phase b Temperature		
10	1.73e-02	
20	8.94e-03	0.95
40	4.55e-03	0.97

(c) beuler, $x_a = 0.70$

N	E_N	r_N
Phase a Temperature		
10	1.59e-03	
20	7.57e-04	1.1
40	3.69e-04	1
Phase b Temperature		
10	2.15e-02	
20	1.03e-02	1.1
40	5.02e-03	1

(d) analytic, $x_a = 0.70$

N	E_N	r_N
Phase a Temperature		
10	4.60e-03	
20	2.83e-03	0.7
40	1.59e-03	0.83
Phase b Temperature		
10	5.06e-03	
20	2.97e-03	0.77
40	1.63e-03	0.86

(e) beuler, $x_a = 0.20$

N	E_N	r_N
Phase a Temperature		
10	2.99e-02	
20	1.53e-02	0.97
40	7.53e-03	1
Phase b Temperature		
10	3.76e-02	
20	1.94e-02	0.96
40	9.57e-03	1

(f) analytic, $x_a = 0.20$

Table 3.29: Numerical error and convergence rates for the coupled drag/conduction tests: temperature results.

3.2.9 Multiphase Particle Cloud

Test Description

This tests the multiphase drag model. It consists of a cloud of solid particles traveling through a gas. The particle cloud is given an initial velocity. As the cloud passes through the stationary gas it experiences a drag force which slows it down. The solid particles account for a very small volume fraction of the overall mixture. Therefore, the flow is fully within the dilute regime and the particles have very little impact on the gas and do not interact with each other. Therefore, ideally each particle within the cloud should behave like an isolated particle. Although the multiphase algorithm does not track each of these particles individually, the bulk motion of the particle cloud should follow the trajectory of an isolated particle.

Test Setup

This problem uses a one-dimensional geometry split into computational cells in the x direction. The length of the domain is one. Initially the domain is divided into three sections: on the left and right is pure gas, in the middle is a section containing a solid-gas mixture representing the particle cloud. The mixture region is initialized with a solid mass fraction of 0.1. This results in a very small volume fraction since the mass density of the solid is 1000.25 while the density of the gas is only 2.5. The particle cloud is initially centered at $x = 0.25$ with a width of 0.1. The particles are given a diameter of 0.0001, their initial velocity is 0.1, and the drag coefficient is assumed to be constant at 1.0,

Test Theory

The theoretical description of a particle in an embedded fluid is discussed in Crowe et al. [4]. The equation of motion for an isolated particle comes from Newton's second law. Applying the standard definition of the drag coefficient gives

$$\frac{\partial^2 x}{\partial t^2} = \frac{\partial v}{\partial t} = -\frac{F_D}{m_p} = -\frac{3\rho_f C_D}{4\rho_p d} v^2 = -cv^2$$

where c is a constant given by

$$c = \frac{3\rho_f C_D}{4\rho_p d}$$

and x and v are the particle position and velocity, respectively. This is a second-order ordinary differential equation. Given the boundary conditions

$$\begin{aligned} v(t=0) &= v_0 \\ x(t=0) &= x_0 \end{aligned}$$

this equation can be integrated to obtain expressions for the particle velocity and position

$$\begin{aligned} v(t) &= \frac{1}{ct + \frac{1}{v_0}} \\ x(t) &= x_0 + \frac{1}{c} \left[\ln \left| ct + \frac{1}{v_0} \right| - \ln \left| \frac{1}{v_0} \right| \right] \end{aligned}$$

The above analytical equations describe the motion of an isolated particle within a carrier gas. Thus, these equations can be applied to each individual particle within a particle cloud. This assumes that the particles within the cloud are spatially separated from each other and do not interact. Therefore, for dilute flow, the above equations can be applied to the velocity and position of the center of mass of a particle cloud.

In order to compare to the theoretical solution, the position and velocity of the center of mass of the particle cloud must be computed at each time-step. The position x_{com} and velocity v_{com} of the center of mass is computed each time-step using:

$$\begin{aligned} x_{com} &= \sum \frac{m_i x_i}{m_{tot}} \\ v_{com} &= \sum \frac{m_i v_i}{m_{tot}} \end{aligned}$$

The above summations are taken over each cell in the problem, where m_i is the particle mass in zone i , x_i is the centroid of zone i , v_i is the particle velocity in zone i , and m_{tot} is the total particle mass in the problem:

$$m_{tot} = \sum m_i$$

Test Results

The results of the simulations agree well with the theoretical solutions. Simulations on four grids were performed using 100, 200, 400, and 800 grid cells. Both the first-order and second-order schemes were used in these calculations. The calculations used the analytical integration model for the drag calculations.

The position of the center of mass of the particle cloud is plotted in Figure 3.91 for the first-order method. The velocity of the cloud is plotted in Figure 3.92. Both variables track the general trend of the isolated particle theory. At the end of the simulation the error in the predicted position is about 0.8% and the error in the velocity is about 8%. This is the case for all grids; mesh resolution does not contribute significantly to the error metrics used on this problem. The second-order scheme does not change the results significantly either, as seen in Figure 3.94 and Figure 3.95.

Some assumptions in the theoretical description of this problem are that all of the particles follow the path of an isolated particle and the particles do not influence the surrounding gas. However, this is not exactly what happens in simulations which treat the particles as a continuum. In the current method, the particles behave like a fluid rather than discrete particles. This has a couple consequences.

First, numerical dissipation causes the particle cloud to diffuse. Thus, the particles will spread out over a larger volume as they advect through the mesh. Plotting the particle volume fraction (Figure 3.93) shows this effect while exhibiting that the numerical diffusion decreases as the mesh is refined. Turning the predictor on reduced this diffusion further, but it still exists as seen in Figure 3.96.

Also, because of the initial gradients in volume fraction, the non-conservative hydrodynamic terms will contribute to the solution. For dilute flow this is expected to be small, but it will result in some error. It is expected that most of the error in this problem comes from these effects. This is why spatial resolution does not improve the solution significantly.

Test Conclusions

The following conclusions can be drawn from this calculation:

- The multiphase model reasonably reproduces position and velocity of isolated particle motion.
- Increasing mesh resolution and turning the predictor on does not improve the center-of-mass calculations, but they do reduce the diffusion of the cloud.
- It is expected that most of the errors associated with the current simulations can be attributed to approximations required to relate the problem setup to the theory.

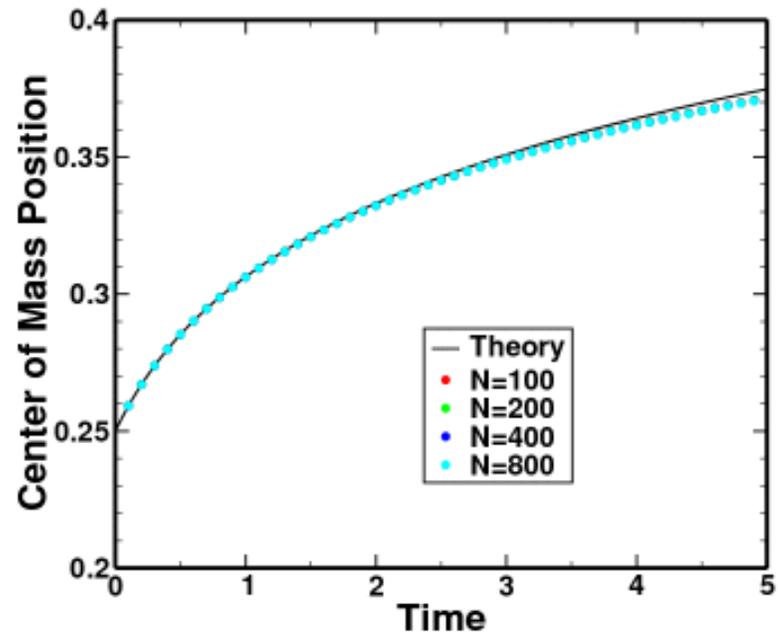


Figure 3.91: Multiphase numerical results for the position of the center of mass of a particle cloud. The tests used an *Eulerian grid* with the *predictor off*.

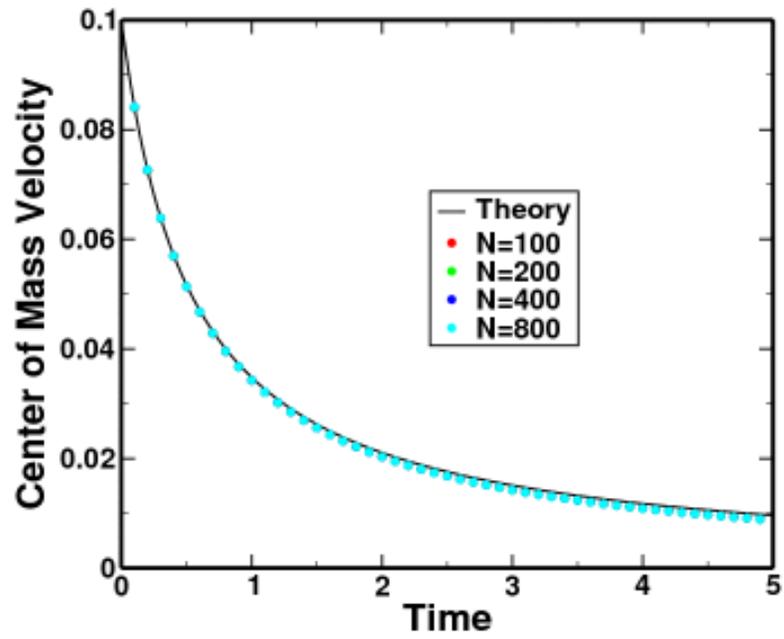


Figure 3.92: Multiphase numerical results for the velocity of the center of mass of a particle cloud. The tests used an *Eulerian grid* with the *predictor off*.

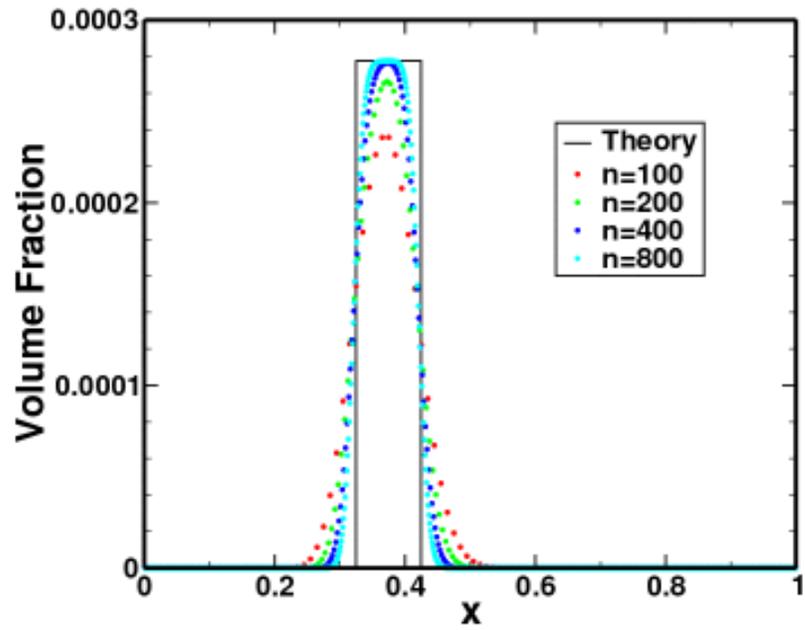


Figure 3.93: Multiphase numerical results for the volume fraction of a particle cloud. The tests used an *Eulerian grid* with the *predictor off*.

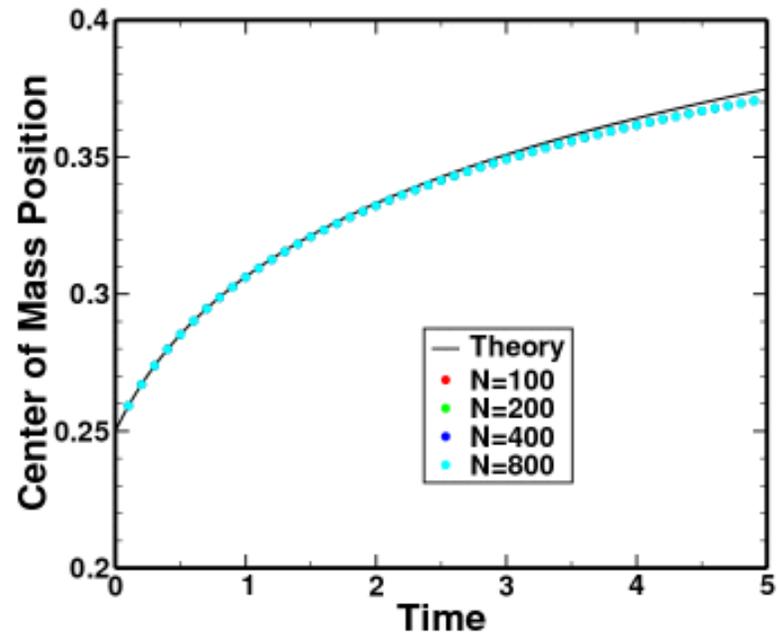


Figure 3.94: Multiphase numerical results for the position of the center of mass of a particle cloud. The tests used an *Eulerian grid* with the *predictor on*.

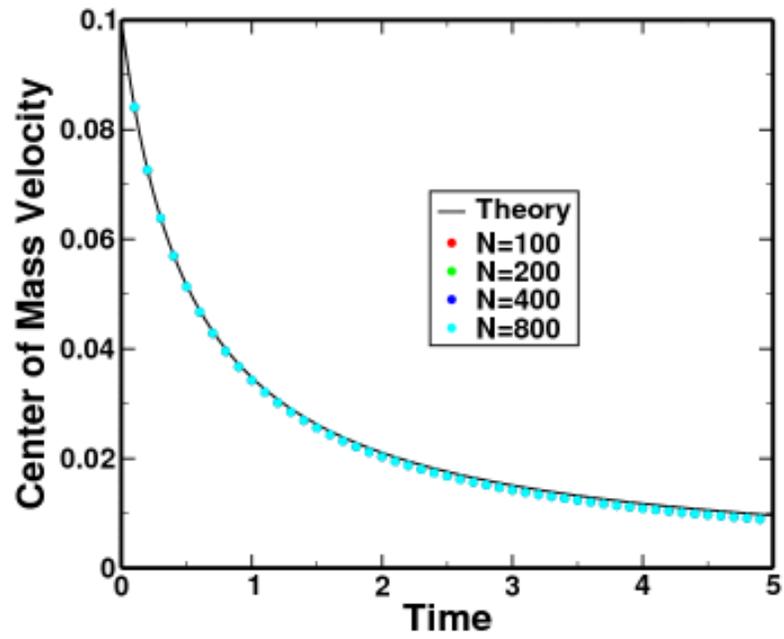


Figure 3.95: Multiphase numerical results for the velocity of the center of mass of a particle cloud. The tests used an *Eulerian grid* with the *predictor on*.

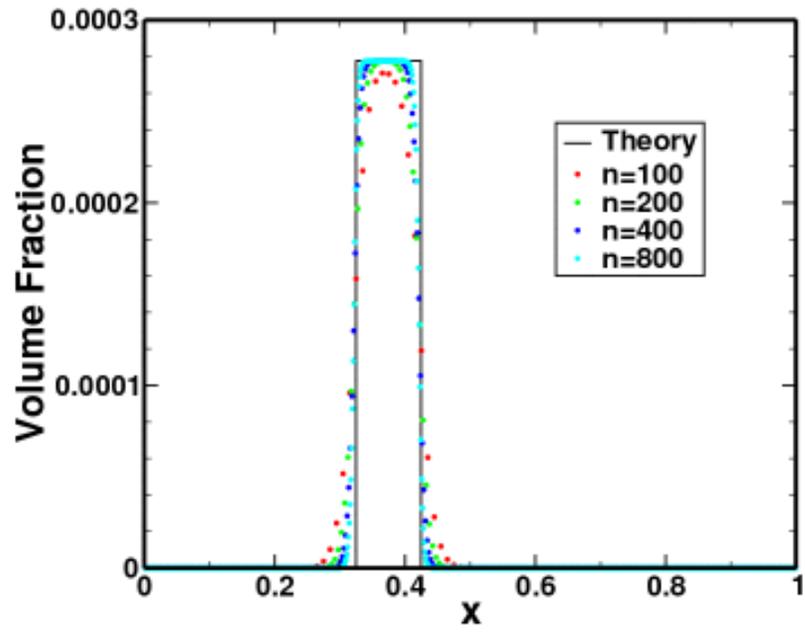


Figure 3.96: Multiphase numerical results for the volume fraction of a particle cloud. The tests used an *Eulerian grid* with the *predictor on*.

3.2.10 Sommerfeld Dusty Shock Tube

Test Description

A planar shock passing through a dusty gas is used to investigate shock attenuation. The problem is initialized with a weak shock in a region of clean air. After propagating a short distance, the shock enters into a very dilute gas-particle mixture. The current results are compared to the experimental data of Sommerfeld [92], who also provided much theoretical analysis of the problem. The problem was previously studied computationally by Sivier et al. [9, 12].

As the shock penetrates the particulate region, a shock is reflected back into the clean air and the original shock is transmitted into the mixture. The particle interactions cause a reduction in the velocity of the transmitted shock. This deceleration does not occur instantaneously due to the inertia of the particles. Thus, a transition region develops where the shock velocity is slowed from its initial speed to the equilibrium value. The experimental data reveal that the resulting shock is strongly influenced by the particle loading and the specific heats of the particles and gas. Accurately modeling the particle drag and heat transfer between phases is crucial to robustly predicting the flow.

Test Setup

This test is setup as a one-dimensional shock tube. Initially there are three sections, labeled A , B , and C in Figure 3.97. On the left (region A) is high pressure air. Next to it in the middle (region B) is low pressure ambient air. Finally, on the right (region C) is a mixture region of air and particles at ambient air pressure. The ambient conditions for this problem are defined as $p = 1 \times 10^{-6}$ Mbar and $T = 300$ K. Given the ambient conditions in front of the shock, the state of the air behind the shock (region A) can be computed using the ideal-gas normal-shock relations [89]. The shock Mach number is 1.49 for the current problem.

The air is modeled using the ideal-gas law with the following properties:

Adiabatic constant

$$\gamma = 1.4$$

Specific heat

$$C_v = 717.5 \times 10^{-8} \frac{Mbar.cm^3}{g.K}$$

Thermal conductivity

$$k = 0.0241 \times 10^{-13} \frac{Mbar.cm^2}{\mu s.K}$$

Coefficient of viscosity

$$\mu = 1.7894 \times 10^{-10} \frac{g}{cm.\mu s}$$

The particulate phase consists of spherical glass particles distributed homogeneously throughout the cloud. The particles have a uniform size, with a diameter of $27 \times 10^{-4} \text{ cm}$ and a density of $2.5 \frac{g}{cm^3}$. The compressibility of the particles is not expected to have a large impact on the solution, so it is assumed that a simplified equation of state is satisfactory for this problem. Therefore, the particles are represented by the stiffened-gas equation of state (see §A.1.2). The properties of the particles are given by the following constants:

Material constants

$$\gamma = 4.4, \quad P_\infty = 0.006 \text{ Mbar}$$

Specific heat

$$C_v = 766.0 \times 10^{-8} \frac{Mbar.cm^3}{g.K}$$

Cold energy

$$\varepsilon_{cold} = 0.000808 \frac{Mbar.cm^3}{g}$$

The particle loading, defined as the mass of the particles divided by the mass of the gas, has a value of 0.63.

Relaxation terms play a very important part in this simulation. Particle drag is given by a drag coefficient which is a function of Reynolds number (Re) as given by the correlation of Schiller and Naumann (see §B)

$$C_D = \frac{24}{Re} (1 + 0.15Re^{0.687})$$

which is valid for $Re < 800$. Calculations in Sommerfeld [92] used an unsteady drag model which was tuned specifically to the current problem.

Heat transfer between the phases is accomplished using the Ranz-Marshall correlation for the Nusselt number (see §C)

$$Nu = 2 + 0.6Re^{1/2}Pr^{1/3}$$

which is a good fit for Reynolds numbers below 50,000. Numerical experiments found that turning off thermal conduction between the phases cuts the shock attenuation in half, while using a constant Nusselt number of 2 over-predicts the attenuation slightly. Thus, heat transfer plays a very important role in this problem.

Due to the very dilute nature of the flow, pressure relaxation is accomplished by slaving the solid pressure to the gas pressure, thus the solid does not influence the gas pressure.

The problem is one dimensional by design. It is modeled using a three-dimensional mesh with a single zone in the y and z directions and symmetry boundary conditions on the sides, resulting in a pseudo-1D flow in the x direction. The number of zones in the x direction are varied to look at mesh convergence. Outflow boundary conditions are given at the left and right boundaries.

Test Theory

No analytic solution is available for this test. Therefore, the results are compared to the experimental data in Figure 15 of Sommerfeld [92].

Test Results

The results are plotted in Figures 3.98–3.101. The experimental data is represented by solid black symbols and the numerical results are given by colored lines. The different colors represent different levels of grid refinement. The grids used in the current results contained 200, 400, and 800 total grid cells. Results using an Eulerian grid are shown in Figure 3.98 with the predictor off, and in Figure 3.99 with the predictor turned on. The results on an ALE mesh are provided in Figure 3.100 and Figure 3.101.

The computations seem to robustly capture the equilibrium shock speed seen in the experiments. However, the computations show a much faster deceleration of the shock than the experiments.

Comparing the predictions using the predictor compared to the answers with no predictor show that the predictor accelerates mesh convergence much faster, as expected. The values obtained with the predictor on show very good results even at very coarse grids, thus indicating that discretization error is very small in this problem. Thus, any improvement in the solution will need to come from improvement in the physics models.

Comparing the fixed grid verses the moving grid results show that the ALE answers seem to have less spread between different resolutions. This implies that the ALE method is moving mesh into the important regions resulting in faster mesh convergence.

Test Conclusions

The following conclusions can be drawn from this calculation:

- The ALE technique allows for improved mesh resolution in the shock region, resulting in improved results compared to an Eulerian calculation using the same number of cells.
- Heat transfer plays a very important role in this problem. Turning off thermal conduction between the phases reduces the shock attenuation by a factor of 2. Therefore, it is assumed that improved results could probably be attained by using a more accurate model for the Nusselt number.
- The drag model also plays an important role in the accuracy of the predictions. Sommerfeld [92] used an unsteady drag model which was tuned specifically to the current problem. It is not clear how well this model would perform when applied to another problem. It is also not clear that an unsteady correction is required for this problem, as the acceleration of the

particles occurs primarily within a very small region around the shock and the fluid acceleration is accounted for by other hydrodynamic terms.

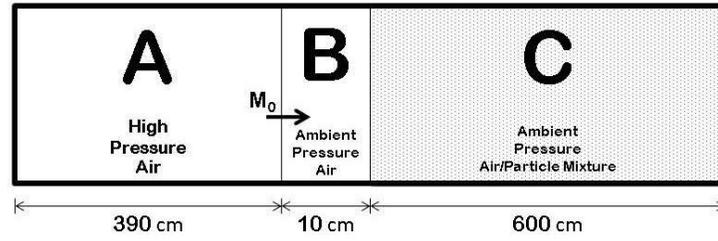


Figure 3.97: Sommerfeld Shock Tube Setup. (Not shown to scale)

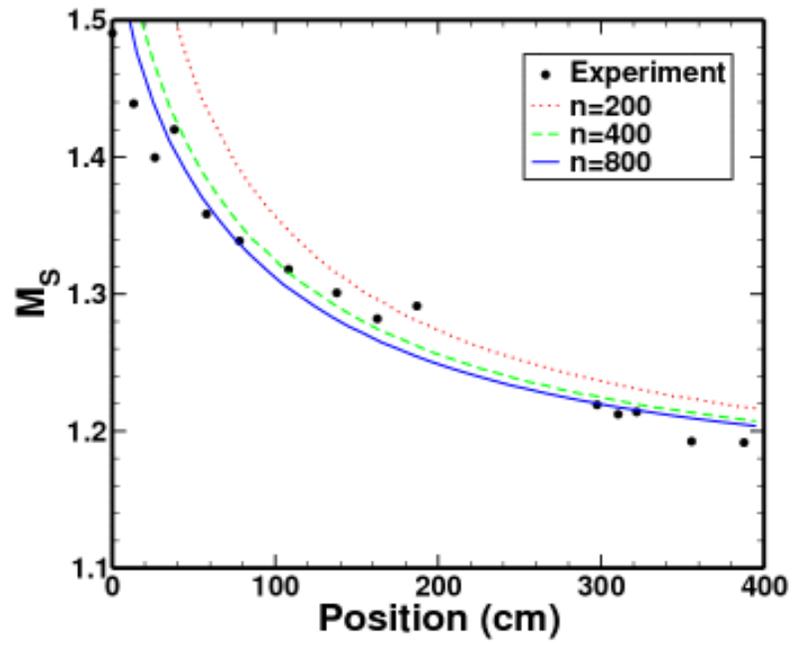


Figure 3.98: Multiphase numerical results for the Sommerfeld shock tube. The shock Mach number is plotted. The tests used an *Eulerian grid* with the *predictor off*.

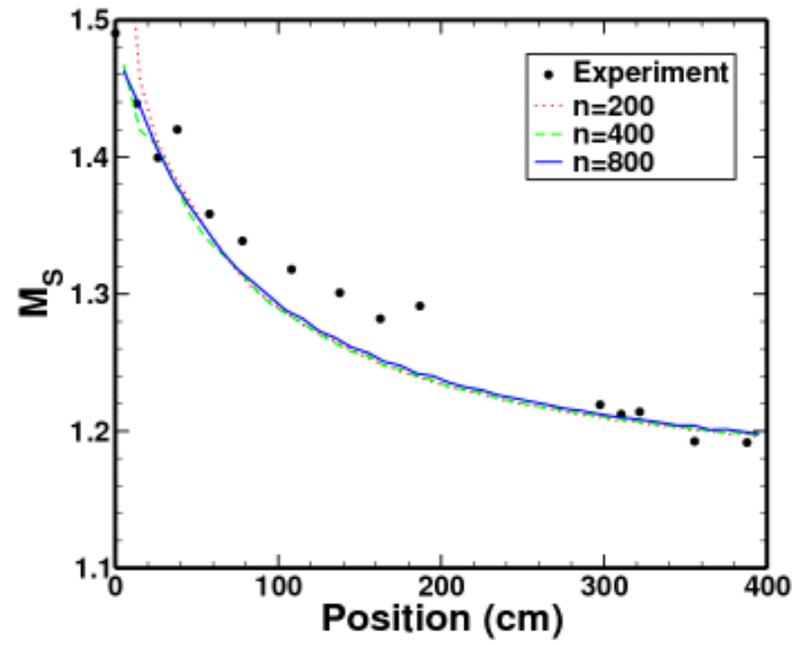


Figure 3.99: Multiphase numerical results for the Sommerfeld shock tube. The shock Mach number is plotted. The tests used an *Eulerian grid* with the *predictor on*.

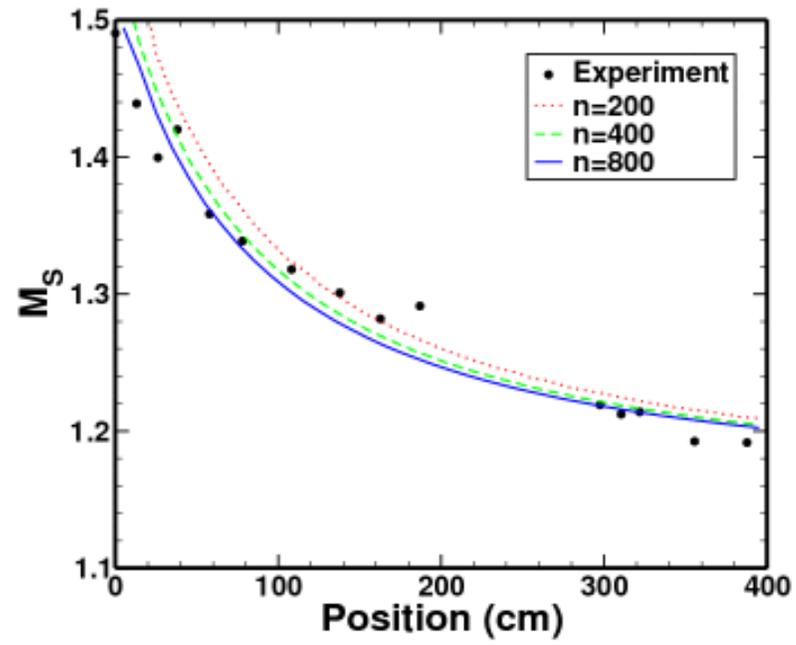


Figure 3.100: Multiphase numerical results for the Sommerfeld shock tube. The shock Mach number is plotted. The tests used an *ALE grid* with the *predictor off*.

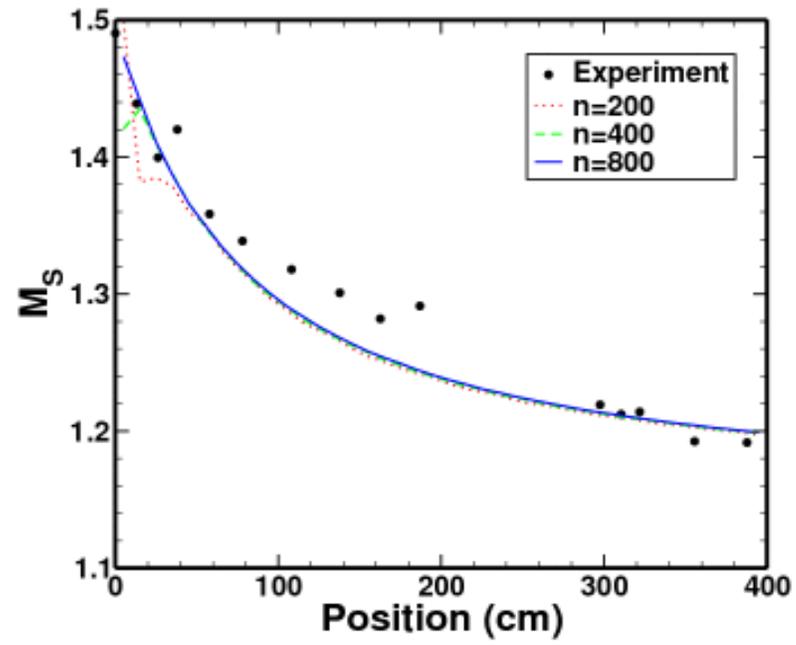


Figure 3.101: Multiphase numerical results for the Sommerfeld shock tube. The shock Mach number is plotted. The tests used an *ALE grid* with the *predictor on*.

3.3 Multiphase-Lagrange Coupling

The final result section describes tests intended to evaluate the interaction between a multiphase fluid and a structural material. This is the ultimate goal of the current effort.

The tests begin with a simple one-dimensional problem involving a multi-phase fluid interacting with a rigid piston. This is followed by a multi-material Riemann problem. The model predictions are next compared to an experimental study of a shock wave impacting a flexible steel plate. Finally, the model is used to investigate the effect of dust particles on a blast wave impacting a structure.

3.3.1 Multiphase Piston

Test Description

This tests the interaction of the multiphase model with a solid. It consists of a multiphase gas on the left, a solid piston in the center, and an ideal gas (staggered-grid hydro) on the right. Initially, the multiphase gas on the left is at a higher pressure than the gas on the right, resulting in a pressure force pushing the piston to the right. As the piston moves, the volume on the left increases causing a decrease in the pressure on the left. At the same time, the volume on the right decreases causing an increase in the pressure on the right. Eventually, the pressure on the right becomes greater than the pressure on the left causing the piston to decelerate and then get pushed to the left. Thus, the pressure force causes the piston to travel back and forth in a cyclic fashion.

Test Setup

The initial geometry for this problem is shown in Figure 3.102. The problem consists of a steel piston separating two gases. The piston is modeled using properties similar to steel, resulting in an essentially rigid material. The gas on the right uses the standard staggered-grid hydrodynamics model. The gas on the left is calculated using the multiphase model and consists of a multiphase mixture of two identical gases. All gases in the problem are modeled as ideal gases with $\gamma = 1.4$.

This problem uses a one-dimensional geometry. It is modeled using a two-dimensional mesh with a single zone in the y direction and symmetry boundary conditions on the sides, resulting in a pseudo-1D flow in the x direction. Symmetry boundary conditions are also given at the boundaries $x = -2$ and $x = 2$. The mesh has a total of 21 zones: 10 zones in the left gas, 10 zones in the right gas, and a single zone in the piston material.

The problem has a domain length of 4 with the origin ($x = 0$) at the center. The piston's length is 1 with a mass also of 1. Initially, the gas on the left has a length of 1 and the right gas has a length of 2. Thus, the initial volume of the gas

on the right is twice the volume of the gas on the left. The mass in both the left and right gas regions are the same, equal to 0.0015. In the multiphase gas, the mass is evenly split between both phases. Therefore, the pressure in the left gas is initially twice that of the right gas.

Test Theory

The motion of the piston is governed by Newton's second law:

$$m_{sol} \frac{d^2x}{dt^2} = A(p_L - p_R)$$

where x is the center of mass of the piston and m_{sol} is the mass of the piston. The pressure in the left and right gas are given by p_L and p_R respectively. The area, A , is the frontal area of the piston, equal to 1 here. The left and right pressure can be related to the gas volume by assuming it follows the isentropic gas law

$$pV^\gamma = p_*V_*^\gamma = const$$

where p_* and V_* are the pressure and volume at equilibrium. This equation only works if the pressure on each side of the piston is in equilibrium. This is a valid assumption if the piston motion is very slow compared to the pressure wave propagation in the gases.

The volume of the left and right gas can easily be obtained from the piston position. Thus, the final governing equation is

$$\frac{d^2x}{dt^2} = \frac{p_*A}{m_{sol}} \left[\left(\frac{V_*}{V_* + Ax} \right)^\gamma - \left(\frac{V_*}{V_* - Ax} \right)^\gamma \right]$$

In the results that follow, this second-order ODE is solved numerically using a forth-order Runge-Kutta solver [93], to obtain the theoretical solution.

Test Results

The results of the simulations agree well with the theoretical solutions. The first-order scheme was used in these calculations, i.e. the predictor was not turned on.

The position of the center of mass of the piston is plotted in Figure 3.103. The velocity of the piston is plotted in Figure 3.104. Both variables are captured by

the calculations very well. The quality of the results indicate that the pressure on each side of the piston is very close to the equilibrium value given by the isentropic gas law. Mesh refinement is not expected to improve the solution to this problem.

Test Conclusions

The following conclusions can be drawn from this calculation:

- The above results show that the force due to the pressure in the multiphase mixture is correctly accelerating the solid piston.
- The theoretical solution of the problem assumes that the multiphase mixture behaves like a gas. Therefore, the current model used for fluid-structure coupling is appropriate for this problem. This is confirmed by these results.

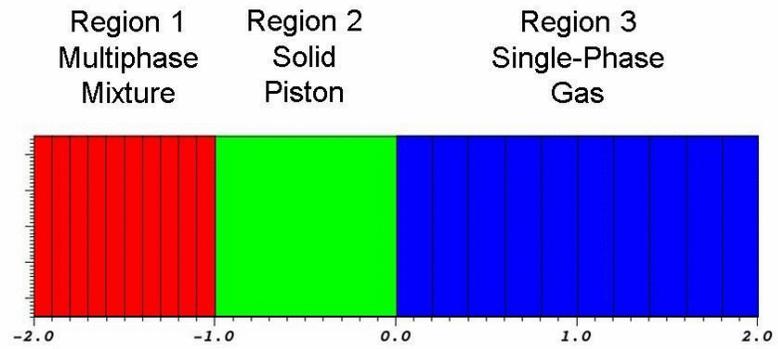


Figure 3.102: Initial geometry for the multiphase piston test. Region 1 is shown in red and contains a multiphase mixture. Region 2 is shown in green and contains a solid material. Region 3 is shown in blue and contains a single-phase gas.

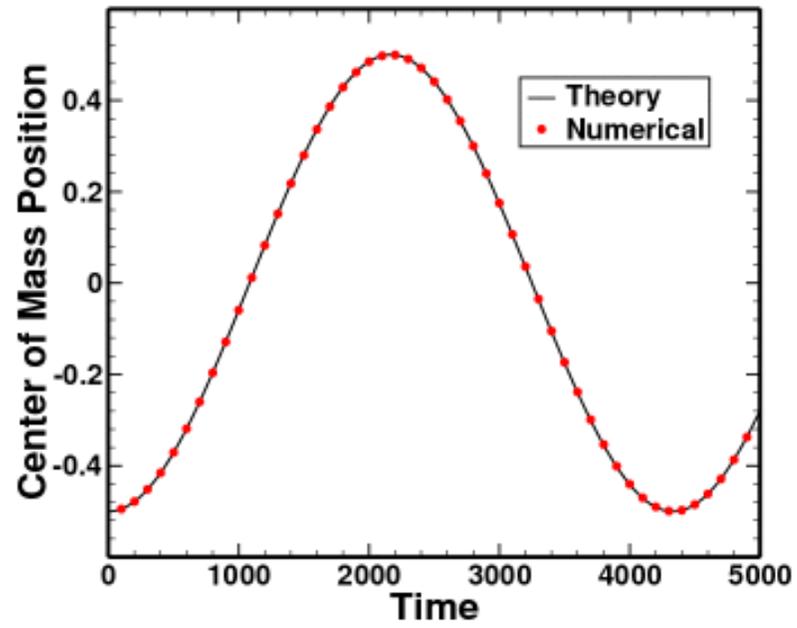


Figure 3.103: Numerical results for the position of the center of mass of a rigid piston.

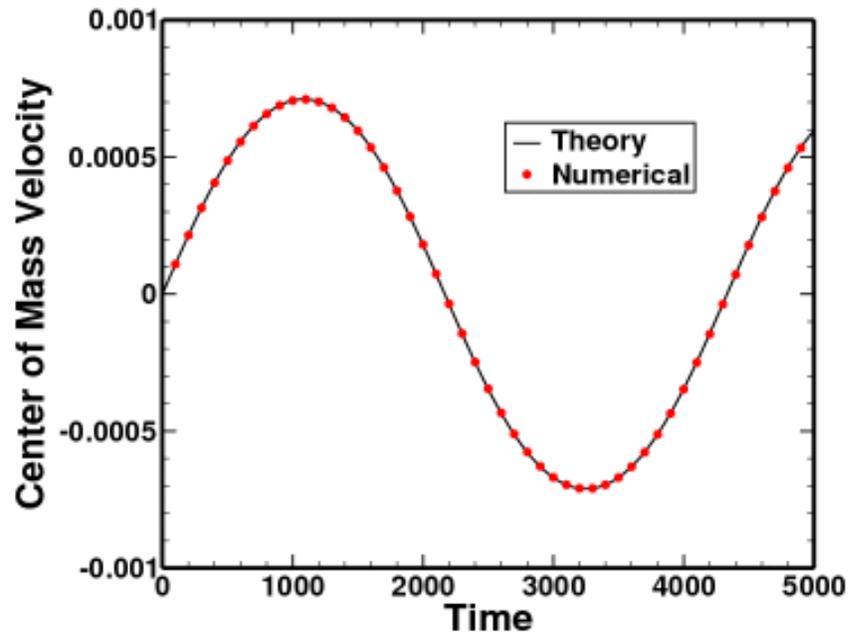


Figure 3.104: Numerical results for the velocity of the center of mass of a rigid piston.

3.3.2 Multiphase-Lagrangian Water-Air Shock Tube

Test Description

A classical one-dimensional shock tube with multiple materials is investigated here. The test consists of high-pressure liquid water on the left and low-pressure air on the right. This has become a standard test for multi-fluid algorithms. It has been modeled using interface tracking methods as well as diffuse boundary methods; including single-phase mixture models [86] and multiphase models [20, 22]. In the current test the water is modeled using the Lagrangian structural model and the air is modeled using the multiphase algorithm.

Test Setup

This test is a one-dimensional simulation of a shock tube with liquid water on the left and air on the right. The problem is similar to the test in §3.2.2, except here the multiphase algorithm is used to model the air on the right, while the Lagrangian structural model is used for the water on the left. In the previous test the multiphase algorithm was used for both materials. Both fluids are represented by the stiffened-gas equation of state (see §A.1.2). The water is represented by $\gamma = 4.4$ and $P_\infty = 6 \times 10^8$ and the air is given by $\gamma = 1.4$ and $P_\infty = 0$. The spatial domain of the problem is $x \in [0, 1]$, where the left and right states are separated by a discontinuity at position $x = 0.7$. The solutions are compared at the simulation end time $t = 229 \times 10^{-6}$. The initial conditions are tabulated in Table 3.30.

The problem is one dimensional by definition. It is modeled using a three-dimensional mesh with a single zone in the y and z directions and symmetry boundary conditions on the sides, resulting in a pseudo-1D flow in the x direction. The number of zones in the x direction are varied to look at mesh convergence. Outflow boundary conditions are given at the boundaries $x = 0$ and $x = 1$.

This test problem is run using a Lagrangian mesh in the water and an ALE mesh in the air. In the air, the mesh motion is determined by the motion of the Lagrangian interface between the materials. As the interface moves to the right and the air is compressed, the air mesh translates to the right keeping a uniform

spacing within the air.

Test Theory

An algorithm for an exact Riemann solver for the stiffened-gas equation of state is presented in [87]. This is extended for multiple materials to obtain the exact solution.

Test Results

The results of the simulations are plotted in Figures 3.105–3.106. The numerical results for the various grid resolutions are given by the colored symbols, while the exact solution is represented by the solid black line. The computations were performed using both the first-order method (predictor off) and the second-order scheme (predictor on) for the multiphase solver. The structural solver used for the water material was always second-order accurate.

This test consists of a rightward traveling shock, a rarefaction wave on the left, and in the middle there is a contact discontinuity traveling to the right following the shock. This is a very severe problem with large pressure and density differences in the materials. The results exhibit the expected numerical diffusion present in shocks and rarefaction waves. The multiphase model appears to capture the shock very well, while the Lagrangian structural model seems to produce overshoots in the density and velocity at the tail of the rarefaction. These overshoots are similar to those seen around rarefactions using the second-order multiphase algorithm in earlier tests. Overall, the propagation speed of the various waves appears to be correct as the average position of the waves is close to the theoretical position. Unlike the pure multiphase version of this problem, the material contact is resolved explicitly in the current approach. Thus, there is no numerical diffusion in the contact discontinuity. The results do exhibit an undershoot in the air density at the material interface. It appears that the solutions are converging to the exact solution as the grid is refined.

Test Conclusions

The following conclusions can be drawn from this calculation:

- The fluid-structure interaction model in the current algorithm adequately computes the quantities in the multi-material problem.
- It appears that the coupling algorithms used in the current approach are appropriate for modeling the interface between two single-phase fluids.

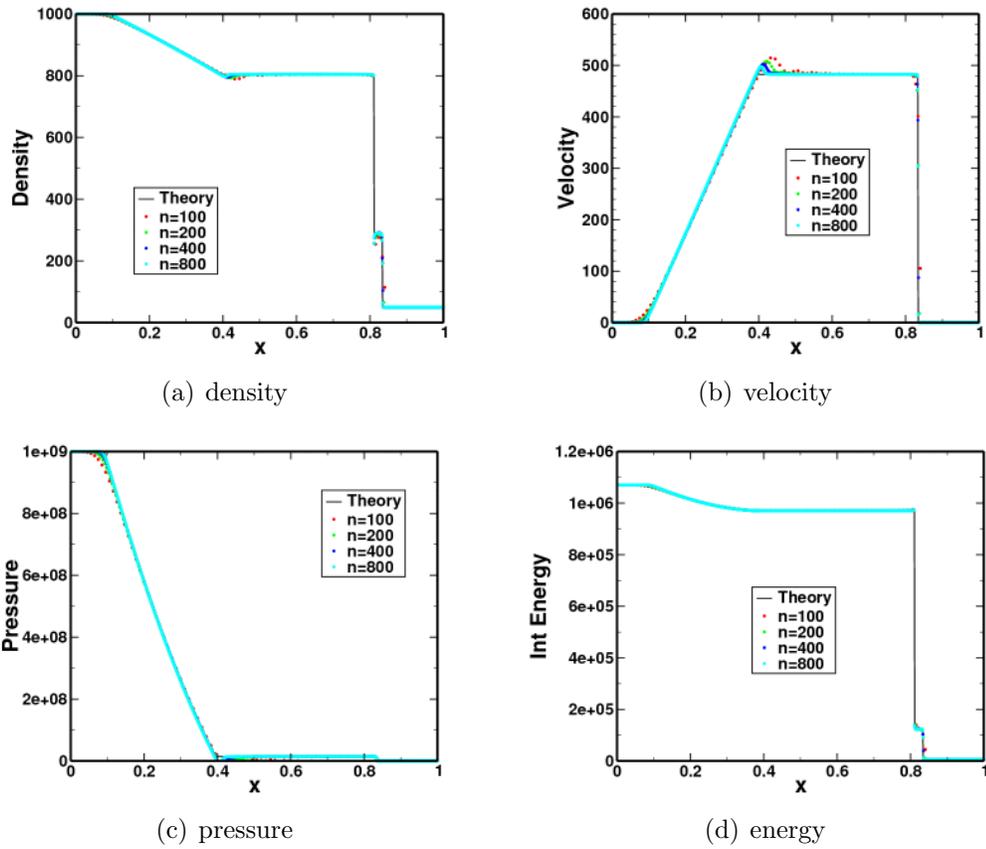


Figure 3.105: Numerical results for the multiphase-Lagrangian water-air shock-tube test. The tests were run with the *predictor off*.

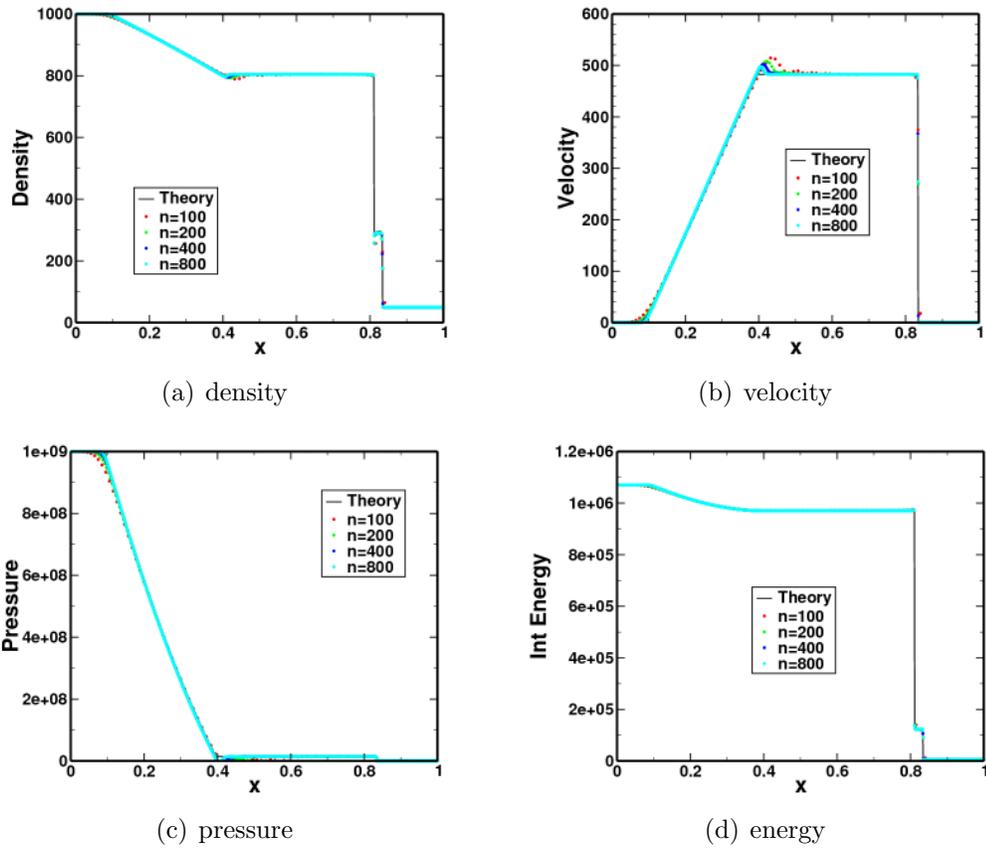


Figure 3.106: Numerical results for the multiphase-Lagrangian water-air shock-tube test. The tests were run with the *predictor on*.

Variable	Value
x_0	0.7
t_0	229×10^{-6}
ρ_L	1000
u_L	0.0
p_L	1×10^9
ρ_R	50
u_R	0.0
p_R	1×10^5

Table 3.30: Initial conditions for the multiphase-Lagrangian water-air shock-tube problem.

3.3.3 Lagrange Cantilever Plate Bending

Test Description

This is a Lagrangian test of the bending of a cantilever plate. This test does not use the multiphase model, it is purely Lagrange hydro. The test is designed to show that the Lagrange hydro can accurately predict the maximum displacement of a plate subjected to a pressure loading (See Figure 3.107). It also computes the fundamental frequency of the plate. This is a prerequisite test for the fluid-structure interaction problem discussed in §3.3.4.

Test Setup

This is a 2D plane-strain problem. The grid is rectangular with a width of 0.1 cm and a height of 5.0 cm. The nodes on the bottom ($y=0$) are constrained.

This problem is run in two stages. The initial stage tests to see if the correct displacement of the top of the plate is calculated. Here the left side has a pressure boundary condition of 2×10^{-6} *Mbar* and the other free surfaces have a pressure boundary condition of 1×10^{-6} *Mbar*.

Once equilibrium is reached, stage two begins. Here the damping is turned off and the pressure on the left side of the plate is reduced to 1×10^{-6} *Mbar* (the same as the other sides). The plate will then begin to oscillate at its fundamental frequency.

The panel is made of an elastic steel. The elastic properties of the steel are given by Giordano et al. [37]. Not given any additional material property data, it was assumed that the Grüneisen equation of state (see §A.1.4) for steel would be adequate. Therefore, the following properties were used:

Reference Density

$$\rho = 7.6 \frac{g}{cm^3}$$

Young's Modulus

$$E = 2.2 \text{ Mbar}$$

Sound Speed

$$c = 0.457 \frac{cm}{\mu s}$$

Bulk Modulus

$$K = \rho c^2 = 1.587 \text{ Mbar}$$

Shear Modulus

$$G = \frac{3KE}{9K - E} = 0.867 \text{ Mbar}$$

Poisson Ratio

$$\nu = \frac{E - 2G}{2G} = 0.269$$

Test Theory

The deflection of an elastic beam with the applied loads shown in Figure 3.107 has an analytical solution. Giordano et al. [37] provides the closed-form solution for the maximum displacement of the end of a beam as well as the fundamental frequency of the beam.

Max Displacement

$$\delta_{max} = \frac{3\Delta p L^4}{2El^3}$$

Circular Frequency

$$\omega = \sqrt{\frac{El^2}{\rho L^4}}$$

Period

$$T = \frac{2\pi}{\omega}$$

The above equations are valid for a long beam which uses the plane-stress approximation. The current problem is the deflection of a long plate. The plate has an infinite depth in the out-of-plane direction. Therefore a plane-strain approximation is more appropriate. The plane-strain problem has the same mathematical form as the plane-stress problem. Therefore, the plane-strain

solution can be obtained from the plane-stress solution by modifying the elastic constants, such that the Young's modulus and Poisson ratio are redefined as

$$\bar{E} = \frac{E}{1 - \nu^2} = 2.37 \text{ Mbar}$$

$$\bar{\nu} = \frac{\nu}{1 - \nu} = 0.368$$

These properties are plugged into the above equations with the following values from the problem setup: $\Delta p = 1 \times 10^{-6} \text{ Mbar}$, $L = 5.0 \text{ cm}$, and $l = 0.1 \text{ cm}$. This yields the result

Max Displacement

$$\delta_{max} = \frac{3\Delta p L^4}{2\bar{E}l^3} = \frac{(3)(1 \times 10^{-6})(5.0)^4}{(2)(2.37)(0.1)^3} = 0.395 \text{ cm}$$

Circular Frequency

$$\omega = \sqrt{\frac{\bar{E}l^2}{\rho L^4}} = \sqrt{\frac{(2.37)(0.1)^2}{(7.6)(5.0)^4}} = 2.23 \times 10^{-3} \frac{\text{rad}}{\mu\text{s}}$$

Period

$$T = \frac{2\pi}{\omega} = 2.81 \times 10^3 \mu\text{s}$$

These results have been tabulated as *Theory* in Table 3.31.

Test Results

The displacement of the plate is plotted in Figure 3.108. The plotted value is the time-history of the x-coordinate of a tracer particle placed at the tip of the plate along the centerline. Three grid-refinement levels were computed. It can be seen that although the solution is not fully grid-converged, the solutions do appear to be converging as the grid is refined.

The first stage of the calculation focused on the maximum displacement of the plate for the given loading. Mass-proportional damping was used in order to reach the equilibrium position more quickly. This stage of the run lasted 10,000 μs , at which point the displacement for each grid resolution was read off the plot

and tabulated in Table 3.31. The error for the finest grid level was about 1.0% compared to the theoretical value.

The next stage of the calculation (10,000 μs to 30,000 μs) focused on the plate oscillation at the fundamental frequency. For this stage, the pressure load was removed and the artificial damping was turned off. The plate displacement went through 6 oscillations in this time. The period of these 6 oscillations were averaged and tabulated for each grid resolution in Table 3.31. The error for the finest grid level was about 0.7% compared to the theoretical value.

Test Conclusions

The following conclusions can be drawn from this calculation:

- The code accurately computed the maximum displacement of the plate deflection.
- The code accurately computed the fundamental frequency of the plate.
- Additional grid refinement could not be done because of excessive run times. The stability limit for the calculations using brick elements is very restrictive. Replacing the brick elements with a shell-element formulation would allow for more efficient simulations.

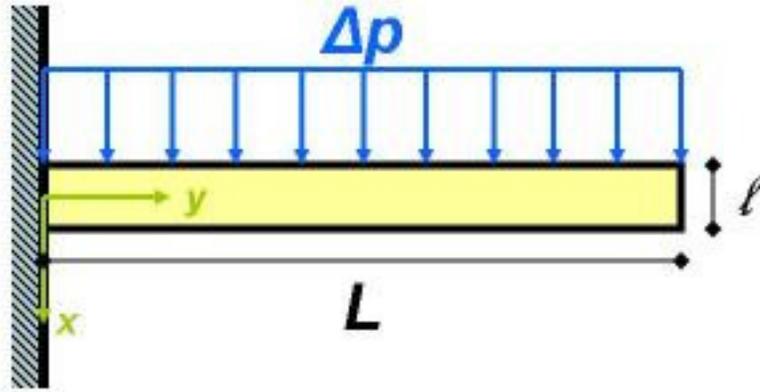


Figure 3.107: Bending-plate problem setup.

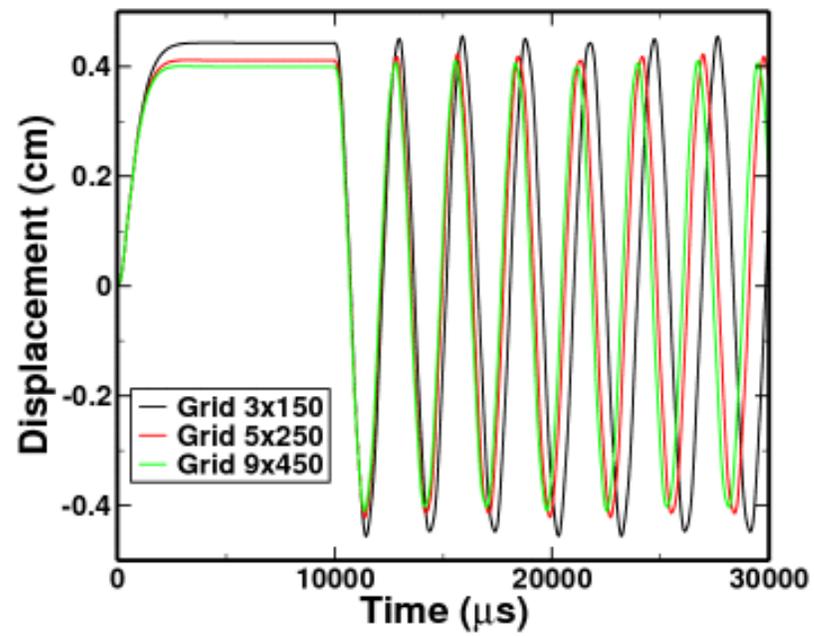


Figure 3.108: Numerical results for the bending-plate problem. Plate displacement is plotted.

	Max Displacement, $\delta_{max}(cm)$	Period, $T(\mu s)$
Theory	0.395	2.81×10^3
3x150	0.443	2.94×10^3
5x250	0.411	2.83×10^3
9x450	0.399	2.79×10^3

Table 3.31: Numerical results for the bending-plate problem. Theoretical and computed plate displacements and periods are tabulated.

3.3.4 Shock Wave Impacting a Steel Panel

Test Description

This is a validation test of the multiphase model for a simulation of a shock wave impacting a steel panel. The problem setup is discussed in Giordano et al. [37]. The test is designed to test fluid-structure interaction for a transient compressible flow. The problem geometry is given in Figure 3.109, which was taken directly from [37]. A thin cantilever plate (the panel) is placed in a shock tube along with its support structure (the base). Both the panel and its base are made of steel. The panel is oriented perpendicular to the flow direction. A pressure jump within the tube generates a weak shock, which impacts the panel and induces flexure of the plate.

Test Setup

A two-dimensional planar geometry was used in the calculations. The problem dimensions are given in Figure 3.109. The domain inlet is located 6 *cm* in front of the center of the panel, and the shock is initially located 3 *cm* in front of the panel. The thickness of the panel is 0.1 *cm*. Data was obtained using four levels of grid refinement. The grids were given the names *x3*, *x5*, *x9*, and *x17*, where the numbers indicate the number of grid cells through the thickness of the panel. The mesh was stretched to allow for more refinement near the panel.

The steel panel and base plate are modeled using the Lagrangian structural model while the air is modeled as a single-phase material using the multiphase method. The steel panel and base plate use a Grüneisen equation of state (see §A.1.4) and an elastic strength model. The steel material properties are the same as those used in §3.3.3. The air uses an ideal-gas equation of state with an adiabatic constant of 1.4. The initial atmospheric conditions in the shock tube are given by $\rho = 1 \times 10^{-3} \text{ g/cc}$ and $p = 1 \times 10^{-6} \text{ Mbar}$. The shock Mach number is 1.21. Given the ambient conditions in front of the shock, the state of the air behind the shock is computed using the ideal-gas normal-shock relations [89].

Test Theory

No analytic solution is available for this test. Therefore, the goal of this test problem is to compare the results with the experimental data found in the original paper by Giordano et al. [37], which also presents numerical calculations (referred to as the GFSI calculation). The experimental data consists of shadowgraphs, panel deflection measurements, and pressure probes.

Test Results

The first data compared with is the flow-visualization images. Giordano presents shadowgraph images of the experiments. Those experimental images are reproduced (on the left) in Figure 3.110. The figure also includes the numerical results of Giordano (in the center) and the current results obtained on grid $x5$ (on the right). The shadowgraph images for the current results were obtained by taking the magnitude of the gradient of the material density. The plot's contour levels and coloring were chosen to closely resemble the experimental images. The shadowgraph plots highlight sharp gradients in the fluid density and therefore the location of shock waves are depicted as dark lines. The panel and base structure also appear as dark objects.

Shadowgraph images are given every 70 microseconds in Figure 3.110. These time levels are based on the experimental data. The time levels of the current results were synchronized such that the initial time level ($t = 0 \mu s$) of the calculation most closely matched the initial time level of the experiments (a few microseconds after the shock impacts the panel). Therefore, the time levels given in the figure are actually offset by about 75 microseconds from the actual calculation.

The current results seem to agree qualitatively with the Giordano experiments and calculations. After the shock interacts with the panel, a reflected shock forms to the left of the panel and a transmitted shock can be seen on the right of the panel. As the flow turns the corner around the panel, a cylindrical shock forms ($t = 70 \mu s$). Note that the current calculations agree very well with the Giordano results for the propagation of the shock wave. Specifically notice that the time

levels at which the shock reflects off the base ($t = 140 \mu s$) and the far right boundary ($t = 630 \mu s$) are similar in all columns of the figure. Even at late time levels, after many complicated shock interactions, the same dominant features of the flow-field in the Giordano results are also predicted in the current calculations.

Figure 3.111 shows the Giordano results and the current calculations on all four of the grids that were run for this problem at $t = 420 \mu s$. As the grid refinement is increased, the shock waves become more distinct in the shadowgraphs. However, the location and structure of the shock waves are very consistent in all grid levels. Therefore, this qualitatively indicates that the fluid solution, at least with respect to the major shock waves, is grid-resolved, even on the coarsest grid shown here. Although not shown here, this is consistent throughout all the time levels.

The vortex that forms off the tip of the panel ($t = 420 \mu s$) in the Giordano results also appears in the current calculations. As the grid is refined, more features become apparent. Eddies form in the finest grids. It is unclear if these eddies are physical or if they are artifacts of the inviscid solver that would go away if viscosity was included in the calculations. The experimental shadowgraphs show a vortex forming off the tip of the panel, but it is less distinct than in the calculations, which may indicate more complex 3D turbulent behavior.

The maximum displacement of the tip of the panel is plotted in Figure 3.112. The experimental results are plotted against the Giordano calculations and the current calculations on the four grid levels. For the current data, the displacement was defined as the x coordinate of the center of the panel. It does not appear that the data in the Giordano results were synchronized exactly to the data in the shadowgraph images. Therefore, in order to match the data in this plot, the time at which the initial panel displacement begins is the synchronization point between the current calculations and the Giordano calculations. This resulted in a zero offset for the current calculations.

The displacement is a good indicator for the adequacy of the grid refinement for the structural solution. Grids $x3$ and $x5$ are both run out to 5 milliseconds.

Due to stability criteria, the finer grids required very small time-steps. Therefore, the solutions on grids $x9$ and $x17$ were not run to completion. However, the data they did provide indicates that the solution is converging as the grid is refined. However, the solution does not appear to be fully grid-converged.

The current calculations appear to reproduce the general trends of the Giordano results, but there are some differences. The experimental results indicate that the maximum displacement of the panel is around 6 *mm*, while the current results suggest it should be closer to 7 *mm*. Also, the period of the oscillation is shorter in the current calculations compared to the experiment. The experimental data also contains more damping of the oscillations than seen in the calculations. The current calculations appear to have the same deficiencies as those in the GFSI calculations. It was postulated that the GFSI deficiencies were caused by deformation of the support structure [37]. In addition it is expected that better material property information would probably improve the agreement.

The experimental test setup used a pressure probe located a short distance in front of the panel on the top of the shock tube. This data is plotted along with the numerical data in Figure 3.113. The current calculations were given a time offset of 45 microseconds to synchronize them to the Giordano results. The calculations agree very well with the experimental data for the initial shock and the subsequent reflection off the panel. After the initial shocks have passed, the calculations slightly overestimate the pressure given in the experiments. Very close agreement is seen between the computations on the four grid levels, indicating once again that the fluid solution is grid-converged.

Test Conclusions

The following conclusions can be drawn from this calculation:

- The multiphase-flow algorithm appears to be coupled to the Lagrangian structure properly for this problem. Comparing the predictions to the experimental data reveal differences, but the trends appear to be captured reasonably.

- Predictions of the flow structure, including shock-wave dynamics, are well correlated to the experimental data.
- The primary differences seen between the simulations and the experiments were in the predictions of the panel deflections. A number of potential causes have been identified for these differences. As discussed in [37], potential errors in the experimental data set are possible due to motion of the support structure. Also, the available material properties are not sufficient to characterize the material response accurately.
- Viscosity was included in the GFSI calculations, but it was not included in the current calculations. Adding viscosity could potentially improve the results slightly, but it is not expected to have large impacts on the solution quality.
- Structural bricks are not the ideal element formulation to use for thin plates. Small mesh spacing within the panel resulted in very small time-step limits, causing excessive run times and preventing additional grid refinement. Thus, switching to a shell-element formulation, as used in the GFSI calculations, would be preferred.

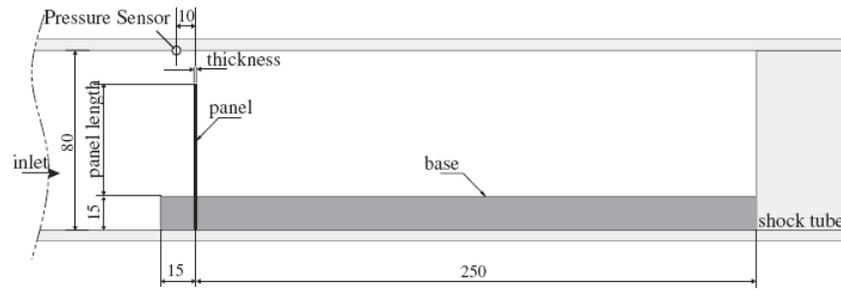


Figure 3.109: Problem setup for a shock wave impacting a steel panel. Figure taken from [37].

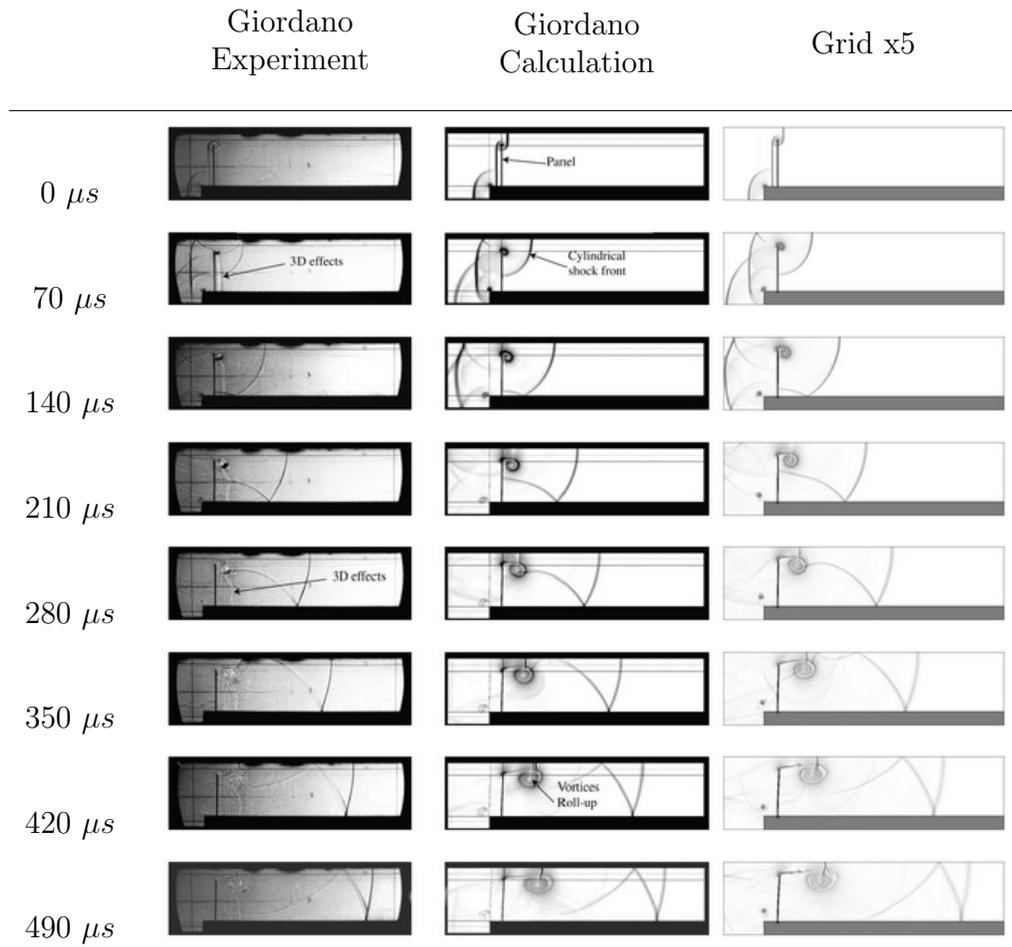


Figure 3.110: Numerical results for a shock wave impacting a steel panel. Experimental and Numerical Shadowgraphs are shown at various time levels. Left column is the experimental shadowgraphs of Giordano. Center column is the numerical calculation of Giordano. Right column is the current calculation on Grid x5.

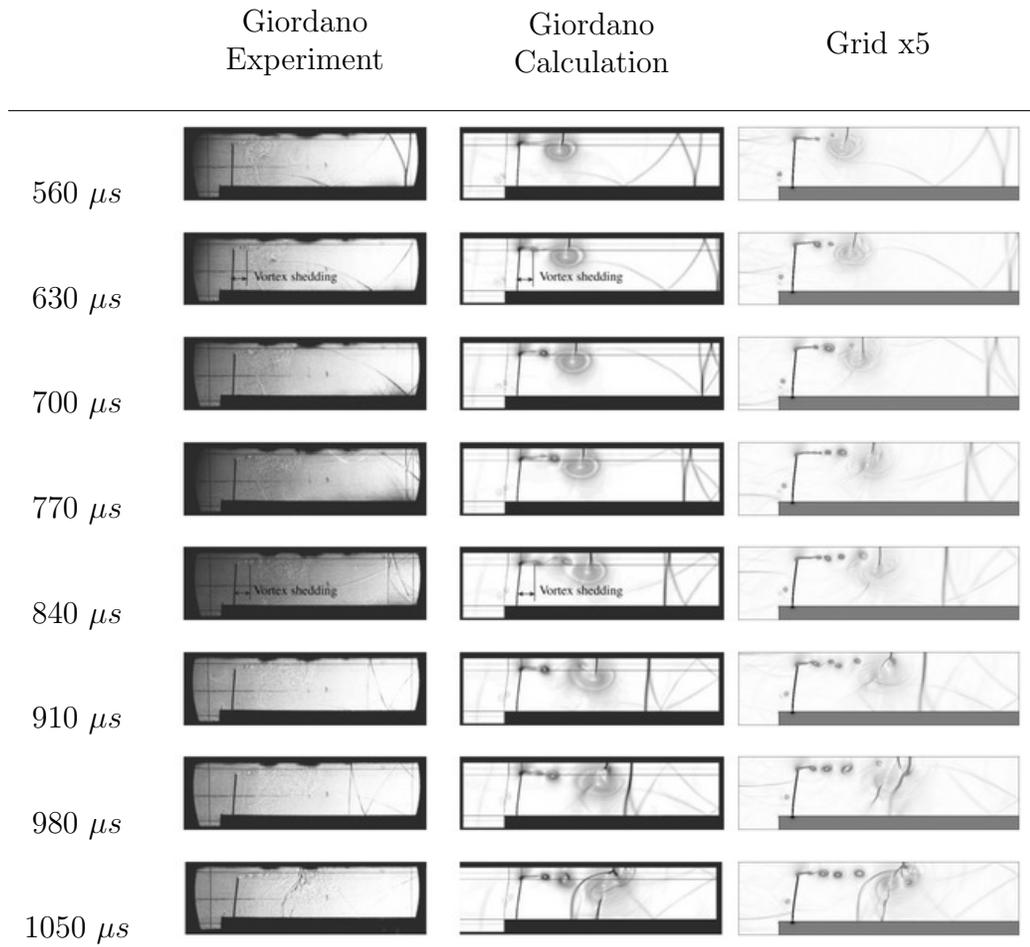


Figure 3.110: Numerical results for a shock wave impacting a steel panel. Experimental and Numerical Shadowgraphs are shown at various time levels. Left column is the experimental shadowgraphs of Giordano. Center column is the numerical calculation of Giordano. Right column is the current calculation on Grid x5. (cont)

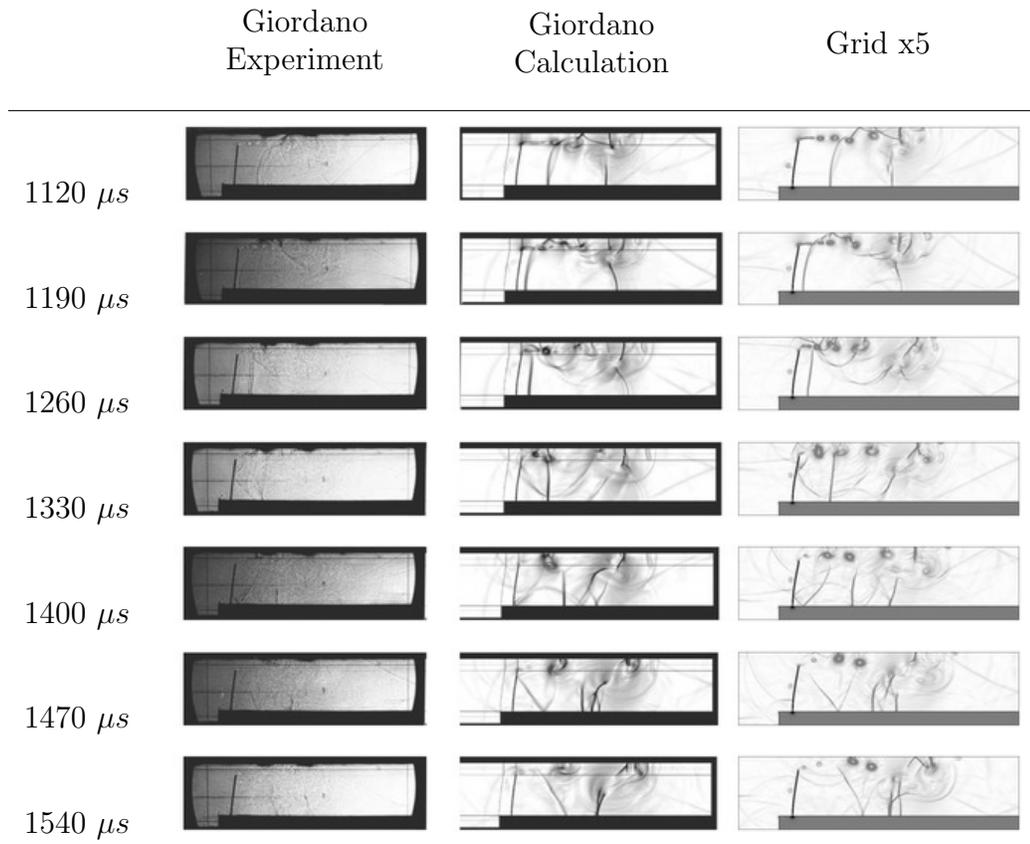


Figure 3.110: Numerical results for a shock wave impacting a steel panel. Experimental and Numerical Shadowgraphs are shown at various time levels. Left column is the experimental shadowgraphs of Giordano. Center column is the numerical calculation of Giordano. Right column is the current calculation on Grid x5. (cont)

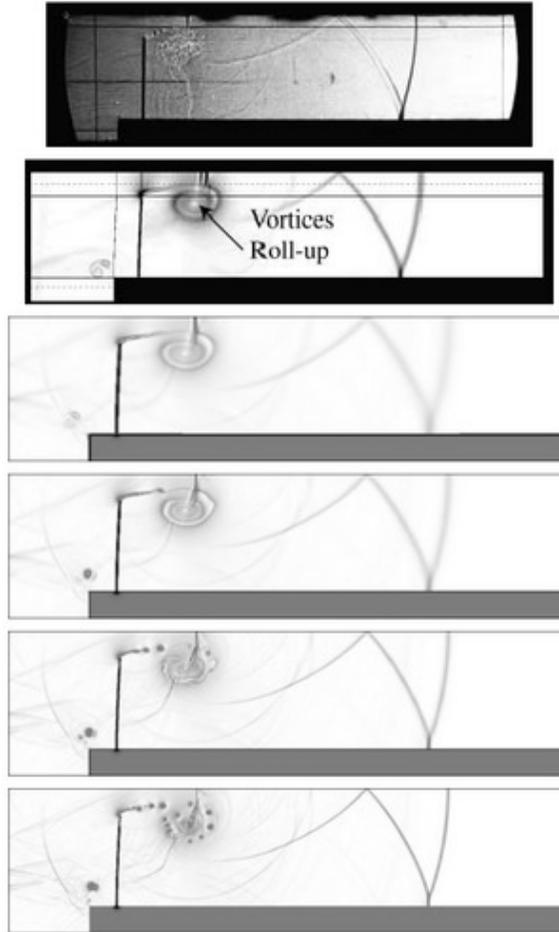


Figure 3.111: Numerical results for a shock wave impacting a steel panel. Shadowgraph taken at $t = 420 \mu s$. From top to bottom: Giordano Experiment, Giordano Calculation, Current Calculation on Grid x3, Grid x5, Grid x9, Grid x17.

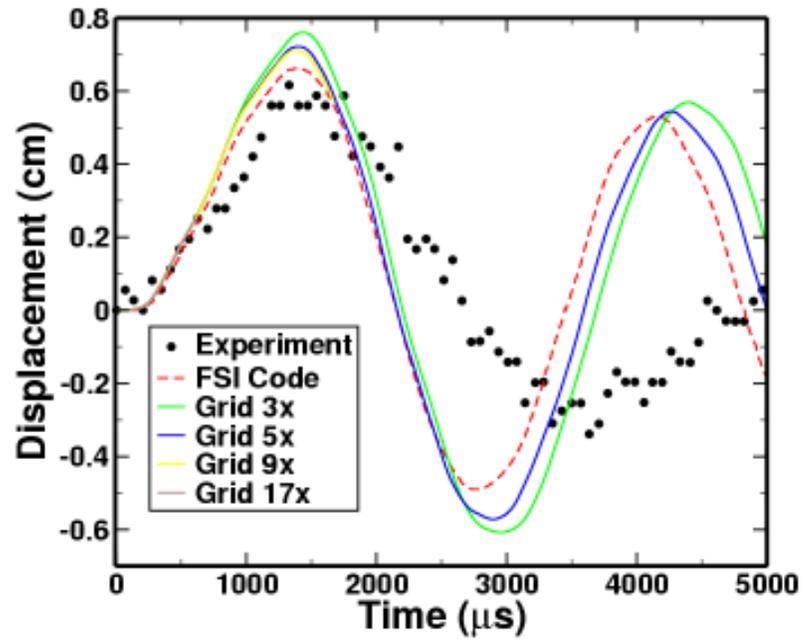


Figure 3.112: Numerical results for a shock wave impacting a steel panel. Plot of panel displacement.

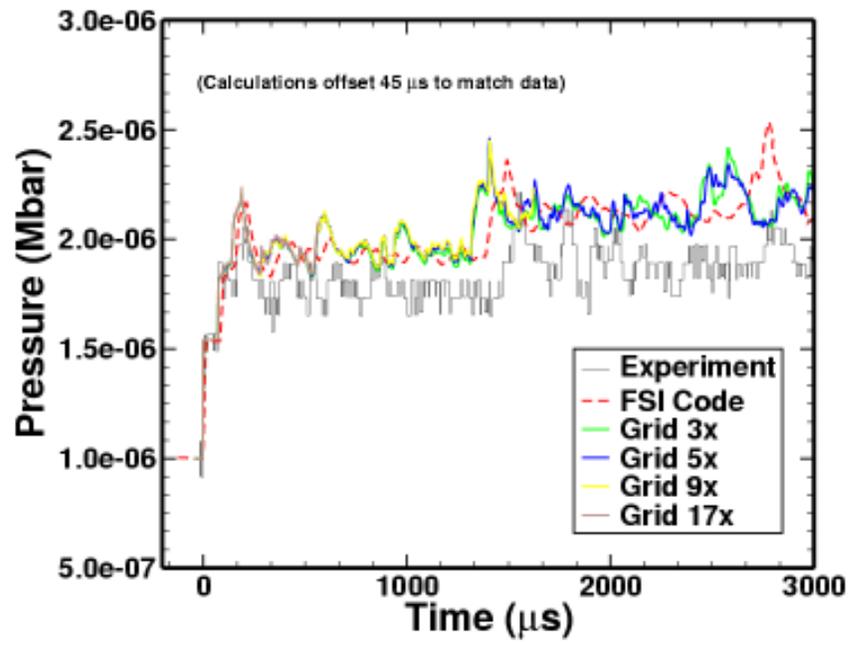


Figure 3.113: Numerical results for a shock wave impacting a steel panel. Plot of pressure probe data.

3.3.5 Multiphase Fluid-Structure Interaction

Test Description

This problem is an exhibition of capability. It consists of a shock in a gas-particle mixture impacting a flexible steel panel. The problem setup is derived from the single-phase fluid-structure interaction problem discussed in the previous section (§3.3.4). A thin cantilever panel made of steel is placed in a shock tube. The panel is embedded within a multiphase gas-particle mixture. This mixture is the same as the one used for the Sommerfeld dusty shock tube (§3.2.10).

This problem is similar to a weak blast wave impacting a structure within a dusty environment. Therefore, this test is analogous to a number of defense and safety applications.

Test Setup

A two-dimensional planar geometry was used in the calculations. The problem dimensions are given in Figure 3.109 and discussed in §3.3.4. Numerical results were obtained using a single level of grid refinement, corresponding to grid $x5$ in the previous study.

The steel panel and base plate are modeled using the Lagrangian structural model while the dusty gas is modeled using the current multiphase method. The material properties of the steel are discussed in §3.3.4. The multiphase mixture consists of air and glass particles. The air is modeled using the ideal-gas law and the particles are represented by the stiffened-gas equation of state. The properties of these materials are given in §3.2.10.

The initial temperature and pressure in the shock tube are given by $T = 300\text{ K}$ and $p = 1 \times 10^{-6}\text{ Mbar}$. The temperature and pressure behind the shock are the same as the ones used for the single-phase test. The problem is initialized with clean air (i.e. no particles) behind the shock at the inlet. To the right of the shock, a particle-air mixture is specified. Therefore, particles initially surround the panel, but there are no particles entrained within the initial shock.

The objective of the current problem is to determine how the presence of

particles in the atmosphere will influence the impulse delivered to a structure from a shock wave. To determine this influence, simulations are performed with various particle loadings. The particle loading is defined as the mass of the particles divided by the mass of the air. For the current calculations, 5 simulations are performed with particle loadings of 0.0, 0.5, 1.0, 2.0, and 9.0. Since the density of the particles is much larger than the density of the air, all of these cases result in a dilute multiphase mixture.

Test Theory

This is an exhibition problem and no analytical theory or experimental data exists for comparison. The goal of this test is simply to exercise the model on a problem of interest.

Test Results

The only metric used in this study is the displacement of the panel. The displacement was defined as the x coordinate of the center of the tip of the panel. This is plotted in Figure 3.114. No experimental data was available for this problem, so only simulation results are shown in the plot. The different colored lines correspond to the initial particle loadings.

As seen in the plot, the panel displacement is reduced as the particle load is increased. This is expected as some of the energy in the shock is used to accelerate the particles, thus the energy transferred to the panel will be less. The particles will also transfer some of their momentum back to the panel due to particle-to-panel collisions. However, due to the inertia of the particles, this effect will lag behind the shock impact. This effect is seen in the results since the period of the panel deflection appears to increase as the particle loading increases, indicating that the duration of the impulsive load on the structure is longer.

To quantify these results, the maximum panel displacement is tabulated in Table 3.32. This table indicates that a particle loading of 9.0 decreases the panel deflection by about 8% compared to the clean (0.0 load) case.

Test Conclusions

The following conclusions can be drawn from this calculation:

- The current multiphase-flow algorithm is capable of simulating a gas-particle mixture impacting a flexible structure. The results show that particles (i.e. dust) around a structure will reduce the deflection of a structure due to a blast wave, while increasing the time duration associated with the impulse. There is no available data to validate the current calculations, but the trends observed in the results seem reasonable.
- No grid convergence study was performed for the current problem. From the results obtained for the single-phase version of the test, the current results are not expected to be grid converged with respect to the displacement predictions. Therefore, additional grid refinement should be performed. However, it is assumed that the trends observed using the current grid will also be present in calculations on a finer mesh.
- It is not known if interactions between the particles and the structure in this problem are handled appropriately. The current model treats the particles like a continuum fluid. Whether or not this is an accurate representation of the physics is an open question. Although the total momentum of the particles is properly conserved, details such as whether the particles bounce or stick to the structure are not included. Therefore, the accuracy of the model will need to be evaluated with experimental data for specific multiphase regimes.

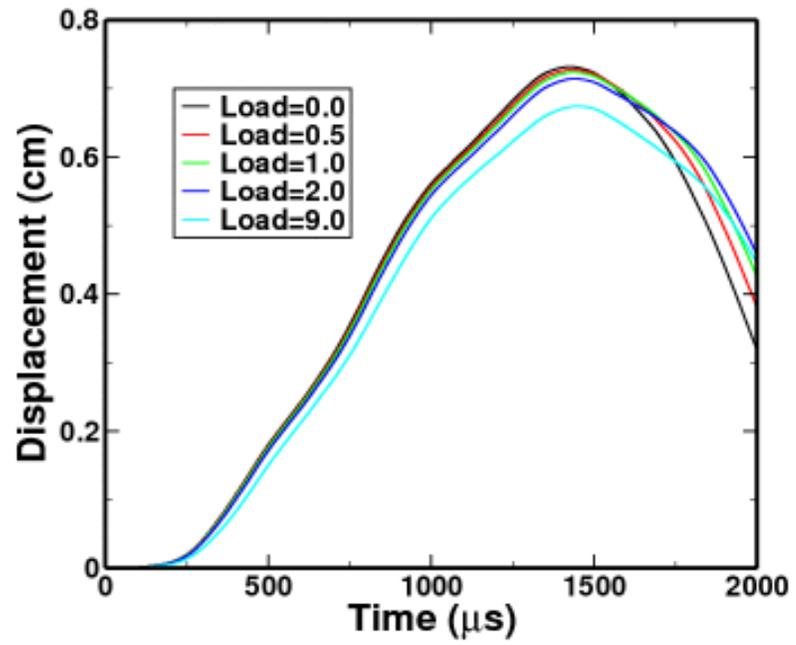


Figure 3.114: Numerical results for a multiphase fluid impacting a steel panel. Plot of panel displacement.

Load	Max Displacement
0.0	0.732
0.5	0.728
1.0	0.724
2.0	0.714
9.0	0.674

Table 3.32: Numerical results for a multiphase fluid impacting a steel panel. Table of maximum panel displacement.

Chapter 4

Conclusions

A novel computational scheme for simulating compressible multiphase flows interacting with solid structures has been developed. The primary application of the current work is the simulation of heterogeneous energetic material, such as explosives or propellants. It is expected that the method will be powerful enough for application to all stages of the problem, including the initial burning of the material, the propagation of blast waves, and interaction with surrounding structures. Although the eventual goal is to utilize the model for reactive materials, the focus so far has been on evaluating the method using inert materials.

The multiphase model has been implemented into an existing hydrocode. Prior to the current project, the code already had a large number of physics models, including an existing Lagrangian structural solver. The code also had a preliminary Eulerian multiphase capability based on the Discrete Equation Method (DEM). This model was rewritten for the current study. The most critical development for the current study was the addition of the ALE capability to run on moving meshes and adding the ability to couple to a Lagrangian structure. This provides the potential to model fluid-structure interaction problems where the working fluid consists of a multiphase mixture. A survey of the literature indicates that this is a unique capability.

The method has been tested on a number of canonical single-phase and multiphase tests. These tests indicate that the scheme is applicable to the

simple materials evaluated and suggest that the scheme would be appropriate for more complicated multiphase problems. To date, the model has primarily been exercised on calculations designed to verify that the model has been implemented correctly and that the governing equations are satisfied. Only a limited number of simulations have been performed to evaluate how well the model predicts particular physics applications, so additional studies are still required. In particular, the empirical relations used in the model, such as drag and heat transfer, must be evaluated with respect to experimental evidence. How well these models represent the behavior of actual gas-particle flows is currently an open question.

The fluid-structure coupling has been tested with some simple verification problems with positive results. The scheme was validated for a single-phase fluid-structure interaction problem with reasonable comparison to experimental data. The method was employed on a similar multiphase fluid-structure interaction problem, although no experimental data is available for comparison. Unfortunately, very little information about relevant experimental studies is found in the literature. Therefore, new experimental studies will need to be initiated. Experiments are primarily needed to evaluate how the discrete phase interacts with the structure. Some open questions that need to be resolved are: Is the momentum properly transferred to the structure? Can a continuum adequately represent individual particles bouncing off or sticking to solid boundaries, or would a Lagrangian representation of the discrete phase be more appropriate? These issues need to be investigated through much experimental validation.

4.1 Future Work

The initial implementation of the multiphase algorithms has been completed for this project. However, this is a long-term effort that will extend beyond the scope of the current study. A number of items have been identified for possible future development.

- A number of enhancements can be made to the current MUSCL scheme.

- It is not currently understood why the second-order scheme fails to demonstrate second-order convergence in smooth multiphase tests (see §3.2.1). A possibility is that the use of the simplified predictor (2.86) is causing the rate of convergence to drop. Perhaps using the full predictor (2.85), including contributions from the multiphase nozzling terms, would regain the desired order of accuracy. Resolving this issue will be an important component of future research.
- The current predictor scheme uses directional splitting to compute the slopes. Historically, this has been the recommended technique. However, Berger et al. [65] recently showed improved results using a gradient-based approach. This method should be evaluated.
- The current scheme interpolates primitive variables to the faces. For multiphase flows, this can result in an inconsistent state. It's possible that the choice of interpolation and limiters could improve performance in this regard, particularly with respect to mass- and volume-fraction positivity. Larrourou [94] discusses an approach which is not directly applicable to the current scheme, but other methods could be applied.
- The method of manufactured solutions is a rigorous technique for order-of-accuracy verification, as discussed by Roache [78] and Salari et al. [79]. The idea is to manufacture an exact solution and then modify the original governing equations by adding source terms in such a way that the manufactured function becomes the exact solution. This could be a very useful way to investigate the numerics of the multiphase algorithm. Applying this method to the DEM is not straightforward due to the discrete nature of the method. Therefore, its implementation will require some research.
- Complications arise due to the evaluation of complex equation of state. Errors are especially common when materials occupy a small volume fraction within a zone. For nearly incompressible materials, such as solids, the EOS is

very stiff and a small error in the volume can result in very large errors in the pressure and temperature. In addition, it is common for the hydrodynamic model to put a phase into a state where the functional form of the EOS is not valid, resulting in unrealistic pressures. One option is to investigate simplified equations of state for the solid phase which are better behaved in the relevant regions. Also, the code uses a number of cutoffs and *ad hoc* adjustments in order to run robustly. These need to be evaluated more fully.

- Pressure relaxation is an open question for the current model. Many multiphase algorithms employ pressure relaxation. The non-conservative terms in the current DEM algorithm provide some mechanical relaxation between the phases. It is not clear whether additional relaxation is appropriate in order to account for sub-grid processes. The various options need to be evaluated for particular flow regimes.
- As discussed, an option to run two-dimensional axisymmetric geometries has been developed. Excellent results are obtained when the multiphase model is run with a single-phase material. When multiple phases are used some asymmetries are observed at the axis. It is possible that the method suffers from the same drawbacks seen in the volume-based scheme discussed in Maire [95] for cell-centered Lagrangian methods. Another possibility is that it is a fundamental issue associated with modeling a discrete phase as a continuum, as seen in Daniel et al. [96]. This should be investigated further.
- A number of problems require an enhancement of the current boundary conditions. Currently, outflow and symmetry conditions are supported. Additional functionality needs to be added.
- Additional dense regime compressible test problems need to be run. This is a critical regime for the simulation of energetic granular materials. These tests will require a compaction model. Therefore, implementation and evaluation of such a model is required.

- The primary focus of this research is in the area of reactive energetic materials. Reactive chemistry models have been implemented in the code. These models have not been discussed in an effort to limit the scope of the current research. These models need to be better evaluated for a wide range of applications.
- Multiphase materials currently interact with the structural-mechanics model through Lagrangian brick elements. However, the current structural-mechanics model has many additional features which do not currently interact with the multiphase model. Structural elements such as shell and beam elements are an example. Slide or contact surfaces are another example. Interaction with these features could be very useful in multiphase calculations and should be incorporated.
- Currently, a Lagrangian interface is required between multiphase and structural material. This limits the amount of deformation allowed by the solid. Relaxing this material boundary is a challenging research area that could be very useful.

Appendix A

Equations of State

This section describes various equations of state used in this work. A thermodynamically consistent equation of state (EOS) model is needed for a correct material description. The models use simplifying approximations in order to capture the dominant physics they aim to capture. Model parameters are fit to match data within a particular regime of interest.

The multiphase model discussed in this work allows for the use of a general EOS. The models are generally functional forms, but in principle any model that returns a pressure and temperature could be used—even tabular data. Since the material density and energy are conserved quantities in the hydrodynamic model, it is natural to express the pressure as the dependent variable, $p = p(\rho, \varepsilon)$. In addition, for problems with thermal effects, a caloric equation of state is required to evaluate the temperature, $T = T(\rho, \varepsilon)$. In general, the temperature obeys the relation

$$T = \frac{1}{C_v} [\varepsilon - \varepsilon_{cold}] \quad (\text{A.1})$$

where C_v is the specific heat and ε_{cold} is the cold energy. The cold energy is a function of density and is defined as the energy at zero temperature or equivalently zero entropy. The EOS must obey the thermodynamic consistency relation

$$\left. \frac{\partial \varepsilon}{\partial v} \right|_T = -p + T \left. \frac{\partial p}{\partial T} \right|_v \quad (\text{A.2})$$

where $v = 1/\rho$ is the relative volume. The cold energy can be determined by

integrating (A.2) along the $T = 0$ isotherm, such that

$$\left. \frac{\partial \varepsilon_{cold}}{\partial v} \right|_T = -p$$

where the pressure is given by its functional form.

A.1 Functional Forms

A.1.1 Ideal Gas

The simplest EOS is the ideal-gas equations of state. It is often referred to as a gamma-law gas. It can be derived from kinetic theory and is most accurate for gases at high temperature and low pressure. As a function of energy and density, the ideal-gas law is expressed as

$$p = (\gamma - 1) \rho \varepsilon \tag{A.3}$$

where the adiabatic index γ is the ratio of specific heats. The temperature is specified as

$$T = \frac{\varepsilon}{C_v}$$

where the cold energy is zero.

A.1.2 Stiffened Gas

The ideal-gas law can be modified for use with high-pressure solids and liquids. This model is called the stiffened-gas or Tammann EOS. Detailed descriptions of this model are found in Menikoff [97] and Le Metayer et al. [98]. Numerical analysis is found in Castro and Toro [99]. The EOS is given by:

$$p = (\gamma - 1) \rho \varepsilon - \gamma P_\infty \tag{A.4}$$

where γ and P_∞ are constant material dependent parameters which can be determined from shock data (see Plohr [100] and Cocchi and Saurel [101] for discussions).

In the above model, the thermodynamically consistent cold energy is given by

$$\varepsilon_{cold} = \frac{P_\infty}{\rho}$$

resulting in the temperature given by

$$T = \frac{1}{C_v} \left[\varepsilon - \frac{P_\infty}{\rho} \right]$$

Although the above model is thermodynamically consistent, it does not always produce a realistic thermal response. Therefore, it is sometimes useful to replace the density dependent cold energy with a constant value, such as

$$\varepsilon_{cold} = \varepsilon_{cold}^0$$

where ε_{cold}^0 is a constant offset used to get a reasonable temperature at the reference state.

A.1.3 JWL

The Jones-Wilkes-Lee (JWL) EOS is commonly used to model detonation product gases. It's form is

$$p = A \left(1 - \frac{\omega\rho}{R_1\rho^0} \right) \exp \left(\frac{-R_1\rho^0}{\rho} \right) + B \left(1 - \frac{\omega\rho}{R_2\rho^0} \right) \exp \left(\frac{-R_2\rho^0}{\rho} \right) + \omega\rho\varepsilon \quad (\text{A.5})$$

where ρ^0 is the reference density for the material. The constants A , B , R_1 , R_2 , and ω are material parameters, see [102]. The consistent temperature relation is

$$T = \frac{1}{C_v} \left[\varepsilon - \left(\frac{A}{R_1\rho^0} \exp \left(\frac{-R_1\rho^0}{\rho} \right) + \frac{B}{R_2\rho^0} \exp \left(\frac{-R_2\rho^0}{\rho} \right) \right) \right]$$

where the exponential terms make up the cold energy.

A.1.4 Grüneisen

The Grüneisen EOS is commonly used for solids. Using cubic shock velocity-particle velocity, its form is given by

$$p = \frac{\rho^0 c^2 \mu \left[1 + \left(1 - \frac{\gamma_0}{2} \right) \mu - \frac{a}{2} \mu^2 \right]}{\left[1 - (S_1 - 1)\mu - S_2 \frac{\mu^2}{(\mu+1)} - S_3 \frac{\mu^3}{(\mu+1)^2} \right]^2} + (\gamma_0 + a\mu)\rho^0\varepsilon \quad (\text{A.6})$$

where ρ^0 is the reference density for the material and μ is the compression given by

$$\mu = \frac{\rho}{\rho_0} - 1$$

The material constants are c , S_1 , S_2 , S_3 , γ_0 , and a . For materials in compression, this equation is reduced to

$$p = \rho^0 c^2 \mu + (\gamma_0 + a\mu)\rho^0 \varepsilon$$

There is no analytical expression available for the cold curve. Therefore, a polynomial fit is used

$$\varepsilon_{cold} = \sum_{i=0}^9 [E_i^c \mu^i]$$

where E_i^c are constants.

Appendix B

Drag Coefficient Relations

This section describes various drag coefficient relations used in this work. The drag coefficient, C_D , is defined by (2.99) in §2.1.14.3. An overview of particulate drag laws can be found in Crowe et al. [4]. Data for the drag around particles is plotted in Figure B.1. The symbols represent experimental data from a number of sources as assembled in White [41]. A subset of the mathematical models discussed below are plotted as solid lines. As seen in the figure, the drag is dependent on the Reynolds number, which is defined as

$$Re = \frac{\rho_c d_p V_R}{\mu_c}$$

where ρ_c is the fluid density, μ_c is the molecular viscosity of the fluid, d_p is the particle diameter, and V_R is the relative velocity between the particle and the surrounding fluid. The data is for an isolated sphere in an incompressible fluid. Drag laws accounting for compressibility have been formulated, but they are not discussed here. For dense particle flows, the porosity of the material must be taken into account. That is done for a number of the following models.

B.1 Newton

The Newton drag law uses a constant value for the drag coefficient,

$$C_D = 0.42$$

The value of the constant defaults to 0.42, but can be modified by the user. The value of 0.42 is reasonable for Reynolds numbers between 10^3 and 10^5 . It is also common to use a value of 2.0 for supersonic flows.

B.2 Stokes

The Stokes drag law is an analytic expression which can be derived assuming Stokes flow where the inertial terms in the Navier-Stokes equations are ignored (see Crowe et al. [4] for details). The Stokes drag coefficient is

$$C_D = \frac{24}{Re}$$

Stokes drag is valid for very low Reynolds numbers ($Re < 1$).

B.3 Schiller and Naumann

A drag relation attributed to Schiller and Naumann (1933) is

$$C_D = \frac{24}{Re} (1 + 0.15Re^{0.687})$$

This relation is an empirical fit to experimental data and is reasonable for $Re < 800$.

B.4 Putnam

A drag relation attributed to Putnam (1961) is

$$C_D = \begin{cases} \frac{24}{Re} (1 + \frac{1}{6}Re^{2/3}) & Re < 1000 \\ 0.4392 & Re \geq 1000 \end{cases}$$

This relation is an empirical fit to experimental data and is reasonable for $Re < 10,000$.

B.5 Ergun

The Ergun (1959) drag law was developed in a study of the pressure drop in a packed bed. It is a combination of the Kozeny-Carman equation and the Burke-Plummer equation. The model is discussed in Crowe et al. [4] and Gidaspow [103].

The associated drag coefficient is

$$C_D = \frac{4}{3} \left(A \frac{(1 - \alpha_g)}{\alpha_g Re} + B \right)$$

where α_g is the volume fraction of the gas. The constants are defined as $A = 150$ and $B = 1.75$, but can be modified by the user.

B.6 RUC

The Representative Unit Cell (RUC) drag model is another model for porous media. It is described in Lundberg and Halvorsen [104]. The model has the same form as the Ergun equation

$$C_D = \frac{4}{3} \left(A \frac{(1 - \alpha_g)}{\alpha_g Re} + B \right)$$

where α_g is the volume fraction of the gas. However, for the RUC model, the values of A and B are mathematically based, such that

$$A = \frac{26.8\alpha_g^3}{(1 - \alpha_g)^{2/3}(1 - (1 - \alpha_g)^{1/3})(1 - (1 - \alpha_g)^{2/3})^2}$$

$$B = \frac{\alpha_g^2}{(1 - (1 - \alpha_g)^{2/3})^2}$$

B.7 Gidaspow

The Gidaspow drag model was developed for packed beds. It is discussed in Gidaspow [103] and Lundberg and Halvorsen [104]. The model is a combination of the Wen and Yu (1966) model for high porosity and the Ergun equation for low porosity. The corresponding drag coefficient for $\alpha_g > 0.8$ is

$$C_D = \alpha_g C_{D0} \alpha_g^{-2.65}$$

where

$$C_{D0} = \begin{cases} \frac{24}{\alpha_g Re} (1 + 0.15(\alpha_g Re)^{0.687}) & Re < 1000 \\ 0.44 & Re \geq 1000 \end{cases}$$

and for $\alpha_g \leq 0.8$

$$C_D = \frac{4}{3} \left(150.0 \frac{(1 - \alpha_g)}{\alpha_g Re} + 1.75 \right)$$

B.8 Akhatov and Vainshtein

The model of Akhatov and Vainshtein [105] uses an interpolation between high and low porosity cases. Defining the parameters

$$C_1 = \frac{24}{Re} + \frac{4.4}{Re^{0.5}} + 0.42$$

$$C_2 = \frac{4}{3\alpha_g} \left(150.0 \frac{(1 - \alpha_g)}{\alpha_g Re} + 1.75 \right)$$

The drag coefficient is

$$C_D = \begin{cases} C_1 & \alpha_g \geq 0.92 \\ C_2 & \alpha_g \leq 0.55 \\ \frac{1}{0.37}(0.92 - \alpha_g)C_2 + (\alpha_g - 0.55)C_1 & 0.55 < \alpha_g < 0.92 \end{cases}$$

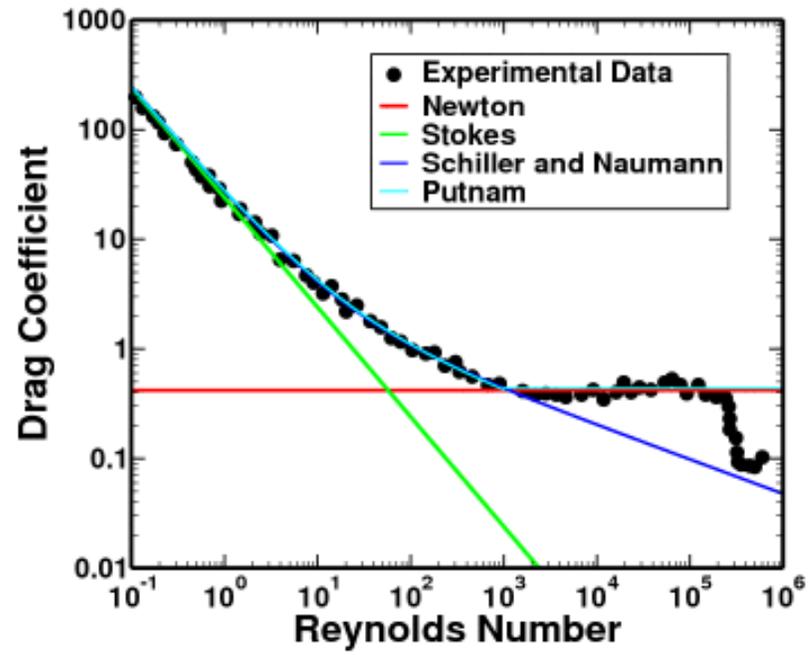


Figure B.1: Comparison of experimental data to empirical relations for the drag coefficient of a smooth solid sphere. Experimental data taken from White [41].

Appendix C

Nusselt-Number Relations

This section describes various Nusselt-number relations used in this work. The Nusselt number, Nu , is defined by (2.92) in §2.1.14.2. An overview of particulate heat transfer can be found in Crowe et al. [4]. The Nusselt-number models allow heat transfer between the phases accounting for conduction and convection. Convection effects are dependent on the properties of the flow, such as Reynolds number and Prandtl number. The Reynolds number is defined as

$$Re = \frac{\rho_c d_p V_R}{\mu_c}$$

where ρ_c is the fluid density, μ_c is the molecular viscosity of the fluid, d_p is the particle diameter, and V_R is the relative velocity between the particle and the surrounding fluid. The Prandtl number is

$$Pr = \frac{\mu_c C_{pc}}{k_c}$$

where k_c is the conductivity of the fluid and C_{pc} is the specific heat at constant pressure for the fluid.

C.1 Constant

The simplest model uses a constant value,

$$Nu = 2.0$$

The value of the constant defaults to 2.0, but can be modified by the user. The value of 2.0 corresponds to heat transfer for a particle in a stagnant medium with no convection effects.

C.2 Ranz-Marshall

The heat-transfer rate increases with flow. The Ranz-Marshall (1951) model accounts for convection effects,

$$Nu = 2.0 + 0.6Re^{1/2}Pr^{1/3}$$

The model is reasonable for Reynolds numbers less than 50,000.

Appendix D

Solution of the Relaxation Equations

This section describes the solution techniques used to solve the thermal- and velocity-relaxation equations derived in §2.1.14.2 and §2.1.14.3. Two distinct solution algorithms have been developed to solve the relaxation ODE's. These are presented first. This is followed with an explanation of how the gas phase is treated in the model.

D.1 Solving the System of ODE's

The goal of the following sections is to solve the following system of non-homogeneous ordinary differential equations:

$$\begin{aligned} \frac{dx_0}{dt} &= -\frac{1}{\tau_0}x_0 + \sum_{i=1}^N \left[\frac{c_i}{\tau_i}x_i \right] - \sum_{i=1}^N [c_i a_i] \\ \frac{dx_i}{dt} &= \frac{1}{\tau_i}x_0 - \frac{1}{\tau_i}x_i + a_i \quad : i = 1, N \end{aligned} \quad (\text{D.1})$$

where c_i , τ_i , a_i are constants and

$$\frac{1}{\tau_0} = \sum_{i=1}^N \frac{c_i}{\tau_i} \quad (\text{D.2})$$

This system is obviously analogous to the equations solved for the thermal (2.96) and velocity (2.104) relaxation modules. In these cases the dependent variable x

simply represents the temperature T or a velocity component of \vec{u} as appropriate. The corresponding constants are also apparent.

In the current code, these equations can be solved using one of two methods. The first is an analytic solution method where a closed-form solution is generated using an eigenvalue decomposition. The second method solves the equations numerically using the backward-Euler time-stepping scheme.

D.1.1 Analytic Relaxation

The system (D.1) can be represented in matrix form by

$$\mathbf{X}' = \mathbf{A}\mathbf{X} + \mathbf{a} \quad (\text{D.3})$$

with initial conditions $\mathbf{X}(t = 0) = \overline{\overline{\mathbf{X}}}$. This system will be solved analytically using an eigenvalue decomposition. First, convert to a symmetric system using the following transformation:

$$\begin{aligned} y_0 &= x_0 \\ y_i &= \sqrt{c_i}x_i \quad : i = 1, N \end{aligned} \quad (\text{D.4})$$

yielding the symmetric system

$$\begin{aligned} \frac{dy_0}{dt} &= -\frac{1}{\tau_0}y_0 + \sum_{i=1}^N \left[\frac{\sqrt{c_i}}{\tau_i}y_i \right] - \sum_{i=1}^N [c_i a_i] \\ \frac{dy_i}{dt} &= \frac{\sqrt{c_i}}{\tau_i}y_0 - \frac{1}{\tau_i}y_i + \sqrt{c_i}a_i \quad : i = 1, N \end{aligned} \quad (\text{D.5})$$

In matrix form this is expressed as

$$\mathbf{Y}' = \mathbf{B}\mathbf{Y} + \mathbf{b} \quad (\text{D.6})$$

with

$$\mathbf{Y} = \begin{bmatrix} x_0 \\ \sqrt{c_1}x_1 \\ \sqrt{c_2}x_2 \\ \vdots \\ \sqrt{c_N}x_N \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} -\frac{1}{\tau_0} & \frac{\sqrt{c_1}}{\tau_1} & \frac{\sqrt{c_2}}{\tau_2} & \dots & \frac{\sqrt{c_N}}{\tau_N} \\ \frac{\sqrt{c_1}}{\tau_1} & -\frac{1}{\tau_1} & 0 & \dots & 0 \\ \frac{\sqrt{c_2}}{\tau_2} & 0 & -\frac{1}{\tau_2} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\sqrt{c_N}}{\tau_N} & 0 & 0 & \dots & -\frac{1}{\tau_N} \end{bmatrix}$$

and

$$\mathbf{b} = \begin{bmatrix} -\sum_{i=1}^N [c_i a_i] \\ \sqrt{c_1} a_1 \\ \sqrt{c_2} a_2 \\ \vdots \\ \sqrt{c_N} a_N \end{bmatrix}$$

To solve the above system, it is desired to diagonalize the coefficient matrix so that the equations become uncoupled. This is done by computing the eigenvalues and eigenvectors. The eigenvalues are computed with the characteristic equation

$$\det(\lambda \mathbf{I} - \mathbf{B}) = \begin{vmatrix} \lambda + \frac{1}{\tau_0} & -\frac{\sqrt{c_1}}{\tau_1} & -\frac{\sqrt{c_2}}{\tau_2} & \dots & -\frac{\sqrt{c_N}}{\tau_N} \\ -\frac{\sqrt{c_1}}{\tau_1} & \lambda + \frac{1}{\tau_1} & 0 & \dots & 0 \\ -\frac{\sqrt{c_2}}{\tau_2} & 0 & \lambda + \frac{1}{\tau_2} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{\sqrt{c_N}}{\tau_N} & 0 & 0 & \dots & \lambda + \frac{1}{\tau_N} \end{vmatrix} = 0$$

which reduces to a polynomial

$$\begin{aligned} \det(\lambda \mathbf{I} - \mathbf{B}) &= \left\{ \prod_{i=1}^N \left[\lambda + \frac{1}{\tau_i} \right] \right\} \left\{ \left(\lambda + \frac{1}{\tau_0} \right) - \sum_{i=1}^N \left[\frac{c_i}{\tau_i^2 \left(\lambda + \frac{1}{\tau_i} \right)} \right] \right\} \\ &= \left\{ \prod_{i=1}^N \left[\lambda + \frac{1}{\tau_i} \right] \right\} \left\{ \lambda + \sum_{i=1}^N \left[\frac{c_i \lambda}{\tau_i \left(\lambda + \frac{1}{\tau_i} \right)} \right] \right\} = 0 \end{aligned}$$

The eigenvalues are the roots of this equation. It is easily seen that one root will always be zero (the right bracket is identically equal to zero if $\lambda = 0$). All other eigenvalues will be negative since all coefficients in the characteristic equation

are positive. The eigenvalue ordering is chosen such that $\lambda_0 = 0$ with successive eigenvalues decreasing in value ($\lambda_i > \lambda_{i+1}$). Once the eigenvalues are known, the eigenvectors can be computed from

$$(\lambda \mathbf{I} - \mathbf{B}) \mathbf{v} = 0$$

where \mathbf{v} is the eigenvector associated with the eigenvalue λ . This system is singular; therefore, it is solved to an arbitrary constant. Looking at rows $i = 1, N$ and assuming there are no duplicate eigenvalues, the i component of \mathbf{v} is computed as

$$v_i = \frac{1}{\bar{v}} \frac{\sqrt{c_i}}{\tau_i (\lambda + \frac{1}{\tau_i})}$$

where \bar{v} is a constant defined in order to create an orthonormal basis. Thus, the eigenvector associated with the eigenvalue λ is

$$\mathbf{v} = \bar{v}^{-1} \begin{bmatrix} 1 \\ \frac{\sqrt{c_1}}{\tau_1 (\lambda + \frac{1}{\tau_1})} \\ \frac{\sqrt{c_2}}{\tau_2 (\lambda + \frac{1}{\tau_2})} \\ \vdots \\ \frac{\sqrt{c_N}}{\tau_N (\lambda + \frac{1}{\tau_N})} \end{bmatrix}$$

where

$$\bar{v} = \left[1 + \sum_{i=1}^N \left[\frac{\sqrt{c_i}}{\tau_i (\lambda + \frac{1}{\tau_i})} \right]^2 \right]^{\frac{1}{2}}$$

For the case with repeated eigenvalues, an alternative strategy must be used to ensure the linear independence of the eigenvectors. Computing the eigenvalues from the characteristic equation is difficult and inefficient for an arbitrary number of phases. Therefore, in practice the eigenvalues and eigenvectors are calculated numerically using LAPACK [106] routines.

After the eigenvectors have been computed, they can be used to diagonalize matrix \mathbf{B} . If the eigenvectors of \mathbf{B} make up the columns of \mathbf{P} , then

$$\mathbf{\Lambda} = \mathbf{P}^{-1} \mathbf{B} \mathbf{P}$$

where the eigenvalues make up the diagonal of $\mathbf{\Lambda}$, such that

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_0 & 0 & 0 & \dots & 0 \\ 0 & \lambda_1 & 0 & \dots & 0 \\ 0 & 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_N \end{bmatrix}$$

For symmetric \mathbf{B} and orthonormal \mathbf{P} , the inverse of the eigenvector matrix is equal to its transpose

$$\mathbf{P}^{-1} = \mathbf{P}^T$$

Thus

$$\mathbf{\Lambda} = \mathbf{P}^T \mathbf{B} \mathbf{P}$$

This can be used to diagonalize the original system of ODE's. Let

$$\mathbf{Y} = \mathbf{P} \mathbf{Z}$$

Then, after substituting this into (D.6), the system can be diagonalized with the following steps:

$$\begin{aligned} \mathbf{Y}' &= \mathbf{B} \mathbf{Y} + \mathbf{b} \\ \mathbf{P} \mathbf{Z}' &= \mathbf{B} \mathbf{P} \mathbf{Z} + \mathbf{b} \\ \mathbf{Z}' &= \mathbf{P}^T \mathbf{B} \mathbf{P} \mathbf{Z} + \mathbf{P}^T \mathbf{b} \\ \mathbf{Z}' &= \mathbf{\Lambda} \mathbf{Z} + \boldsymbol{\gamma} \end{aligned} \tag{D.7}$$

where the non-homogeneous constants are computed as

$$\begin{aligned} \boldsymbol{\gamma} &= \mathbf{P}^T \mathbf{b} \\ \gamma_i &= \sum_{j=0}^N [P_{ji} b_j] = P_{0i} b_0 + \sum_{j=1}^N [P_{ji} b_j] \\ &= P_{0i} \left(- \sum_{j=1}^N [c_j a_j] \right) + \sum_{j=1}^N [P_{ji} (\sqrt{c_j} a_j)] \\ &= \sum_{j=1}^N [\sqrt{c_j} a_j (P_{ji} - \sqrt{c_j} P_{0i})] \end{aligned}$$

Note that for the special case when there are no repeated eigenvalues, the non-homogeneous coefficient associated with the zeroth equation is zero ($\gamma_0 = 0$); although this is not true for the general case.

System (D.7) represents $N+1$ non-homogeneous ordinary differential equations. Since the matrix is diagonal, the equations are uncoupled and can be solved individually. The solutions are of the form

$$z_i(t) = z_i^H(t) + z_i^P(t) \quad : i = 0, N$$

with z_i^H representing the homogeneous solution and z_i^P representing the particular solution. Depending on the value of λ , the solutions can be determined up to an arbitrary constant α_i .

for $\lambda = 0$:

$$\begin{aligned} z_i^H(t) &= \alpha_i \\ z_i^P(t) &= \int \gamma_i dt = \gamma_i t \end{aligned}$$

for $\lambda \neq 0$:

$$\begin{aligned} z_i^H(t) &= \alpha_i e^{\lambda_i t} \\ z_i^P(t) &= e^{\lambda_i t} \int e^{-\lambda_i t} \gamma_i dt = -\frac{\gamma_i}{\lambda_i} \end{aligned}$$

The above solution can be generalized as

$$z_i(t) = \alpha_i e^{\lambda_i t} + g_i(t)$$

where

$$g_i(t) = \begin{cases} +\gamma_i t & \text{if } \lambda_i = 0 \\ -\frac{\gamma_i}{\lambda_i} & \text{if } \lambda_i \neq 0 \end{cases}$$

The arbitrary constants α_i in the solution are determined from the initial conditions. Evaluating the above equation at time zero gives

$$z_i(t=0) = \alpha_i + g_i(t=0)$$

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix} = \mathbf{P}^T \begin{bmatrix} \bar{\bar{X}}_0 \\ \sqrt{c_1} \bar{\bar{X}}_1 \\ \sqrt{c_2} \bar{\bar{X}}_2 \\ \vdots \\ \sqrt{c_N} \bar{\bar{X}}_N \end{bmatrix} - \begin{bmatrix} \bar{\bar{g}}_0 \\ \bar{\bar{g}}_1 \\ \bar{\bar{g}}_2 \\ \vdots \\ \bar{\bar{g}}_N \end{bmatrix}$$

where

$$\bar{\bar{g}}_i = g_i(t=0) = \begin{cases} 0 & \text{if } \lambda_i = 0 \\ -\frac{\gamma_i}{\lambda_i} & \text{if } \lambda_i \neq 0 \end{cases}$$

Therefore, the constants can be computed as

$$\begin{aligned} \alpha_i &= P_{0i} \bar{\bar{X}}_0 + \sum_{j=1}^N [P_{ji} \sqrt{c_j} \bar{\bar{X}}_j] - \bar{\bar{g}}_i \\ &= \tilde{\alpha}_i - \bar{\bar{g}}_i \end{aligned}$$

where the homogeneous component has been split out, such that

$$\tilde{\alpha}_i = P_{0i} \bar{\bar{X}}_0 + \sum_{j=1}^N [P_{ji} \sqrt{c_j} \bar{\bar{X}}_j]$$

It is now possible to finish the solution. Recall,

$$\mathbf{Y} = \mathbf{PZ}$$

Thus, plugging the appropriate values into the solution, gives for $i = 0, N$

$$\begin{aligned} y_i(t) &= \sum_{j=0}^N [P_{ij} (\alpha_j e^{\lambda_j t} + g_j)] \\ &= \sum_{j=0}^N [P_{ij} ((\tilde{\alpha}_j - \bar{\bar{g}}_j) e^{\lambda_j t} + g_j)] \\ &= \sum_{j=0}^N [P_{ij} \tilde{\alpha}_j e^{\lambda_j t}] - \sum_{j=0}^N [P_{ij} (\bar{\bar{g}}_j e^{\lambda_j t} - g_j)] \\ &= \sum_{j=0}^N [P_{ij} (\tilde{\alpha}_j e^{\lambda_j t} + f_j)] \end{aligned}$$

where

$$f_i = \begin{cases} \gamma_i t & \text{if } \lambda_i = 0 \\ \frac{\gamma_i}{\lambda_i} (e^{\lambda_i t} - 1) & \text{if } \lambda_i \neq 0 \end{cases}$$

results from the non-homogeneous part of the original system. The final answer is achieved by a simple transformation back to \mathbf{X} . Thus,

$$\begin{aligned} x_0 &= y_0 \\ x_i &= \frac{1}{\sqrt{c_i}} y_i \quad : i = 1, N \end{aligned}$$

Single Particle Phase ($N = 1$)

For the common case where there is a single particle phase embedded within a fluid, the analytical solution becomes much simpler to express. A closed form solution for the eigenvalues can be determined, such that

$$\begin{aligned} \lambda_0 &= 0 \\ \lambda_1 &= -\beta \end{aligned}$$

where

$$\beta = \frac{c_1 + 1}{\tau_1}$$

Therefore, the resulting eigenvector matrix is

$$\mathbf{P} = \frac{1}{\sqrt{c_1 + 1}} \begin{bmatrix} 1 & \sqrt{c_1} \\ \sqrt{c_1} & -1 \end{bmatrix}$$

Hence, plugging these values into the solution results in the final closed form solution to the original ODE system.

$$\begin{aligned} x_0 &= \bar{\bar{X}}_0 + \frac{c_1}{c_1 + 1} (\bar{\bar{X}}_0 - \bar{\bar{X}}_1) (e^{-\beta t} - 1) + \frac{c_1 a_1 \tau_1}{c_1 + 1} (e^{-\beta t} - 1) \\ x_1 &= \bar{\bar{X}}_1 - \frac{1}{c_1 + 1} (\bar{\bar{X}}_0 - \bar{\bar{X}}_1) (e^{-\beta t} - 1) - \frac{a_1 \tau_1}{c_1 + 1} (e^{-\beta t} - 1) \end{aligned}$$

As it is much more efficient, the code uses the above expressions, bypassing the full eigenvalue solution, for problems with a single particle phase.

D.1.2 Backward-Euler Relaxation

This section discusses backward-Euler relaxation. The backward-Euler method is a numerical integration scheme which is implicit and first-order accurate. It is discussed in many numerical method texts, e.g. [93]. The idea is to solve the ODE system (D.1), which is represented in matrix form as

$$\mathbf{X}' = \mathbf{A}\mathbf{X} + \mathbf{a}$$

Applying the backward-Euler operator to the above system results in the following discretization:

$$\frac{\mathbf{X}^{n+1} - \mathbf{X}^n}{\Delta t} = \mathbf{A}\mathbf{X}^{n+1} + \mathbf{a}$$

assuming that \mathbf{A} and \mathbf{a} are constant. After rearranging, one arrives at a system of linear equations.

$$(\mathbf{I} - \Delta t \mathbf{A}) \mathbf{X}^{n+1} = \mathbf{X}^n + \Delta t \mathbf{a}$$

or

$$\mathbf{B}\mathbf{X}^{n+1} = \mathbf{R} \tag{D.8}$$

The coefficient matrix is defined as

$$\mathbf{B} = (\mathbf{I} - \Delta t \mathbf{A}) = \begin{bmatrix} (\gamma_0 + 1) & -c_1 \gamma_1 & -c_2 \gamma_2 & \dots & -c_N \gamma_N \\ -\gamma_1 & (\gamma_1 + 1) & 0 & \dots & 0 \\ -\gamma_2 & 0 & (\gamma_2 + 1) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\gamma_N & 0 & 0 & \dots & (\gamma_N + 1) \end{bmatrix}$$

where

$$\begin{aligned} \gamma_0 &= \frac{\Delta t}{\tau_0} \\ \gamma_i &= \frac{\Delta t}{\tau_i} \end{aligned}$$

and the right hand side is given by

$$\mathbf{R} = \mathbf{X}^n + \Delta t \mathbf{a} = \begin{bmatrix} x_0^n - \Delta t \sum_{i=1}^N [c_i a_i] \\ x_1^n + \Delta t a_1 \\ x_2^n + \Delta t a_2 \\ \vdots \\ x_N^n + \Delta t a_N \end{bmatrix}$$

Many options are available for solving (D.8). The approach taken here is to apply Cramer's rule [107] since, as will be seen, it lends itself to a closed form solution for matrix \mathbf{B} . According to Cramer's rule, the solution x_i^{n+1} is given by the ratio

$$x_i^{n+1} = \frac{\det(\mathbf{B}_i)}{\det(\mathbf{B})}$$

where \mathbf{B}_i is the the matrix \mathbf{B} with the i column replaced by the column vector \mathbf{R} . For example, for the zeroth column

$$\mathbf{B}_0 = \begin{bmatrix} R_0 & -c_1 \gamma_1 & -c_2 \gamma_2 & \dots & -c_N \gamma_N \\ R_1 & (\gamma_1 + 1) & 0 & \dots & 0 \\ R_2 & 0 & (\gamma_2 + 1) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_3 & 0 & 0 & \dots & (\gamma_N + 1) \end{bmatrix}$$

Therefore, the determinant in the numerator can be evaluated by co-factor

expansion to obtain

$$\begin{aligned}
\det(\mathbf{B}_0) &= \begin{vmatrix} R_0 & -c_1\gamma_1 & -c_2\gamma_2 & \dots & -c_N\gamma_N \\ R_1 & (\gamma_1 + 1) & 0 & \dots & 0 \\ R_2 & 0 & (\gamma_2 + 1) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_3 & 0 & 0 & \dots & (\gamma_N + 1) \end{vmatrix} \\
&= \left\{ \prod_{j=1}^N [\gamma_j + 1] \right\} \left\{ R_0 + \sum_{j=1}^N \left[\frac{R_j c_j \gamma_j}{(\gamma_j + 1)} \right] \right\} \\
&= \left\{ \prod_{j=1}^N [\gamma_j + 1] \right\} \left\{ x_0^n - \Delta t \sum_{j=1}^N [c_j a_j] + \sum_{j=1}^N \left[\frac{c_j \gamma_j}{(\gamma_j + 1)} (x_j^n + \Delta t a_j) \right] \right\} \\
&= \left\{ \prod_{j=1}^N [\gamma_j + 1] \right\} \left\{ x_0^n + \sum_{j=1}^N \left[\frac{c_j \gamma_j}{(\gamma_j + 1)} x_j^n - \frac{c_j a_j \Delta t}{(\gamma_j + 1)} \right] \right\} \\
&= \left\{ \prod_{j=1}^N [\gamma_j + 1] \right\} \left\{ x_0^n + \sum_{j=1}^N \left[\frac{c_j \gamma_j}{(\gamma_j + 1)} (x_j^n - a_j \tau_j) \right] \right\}
\end{aligned}$$

Similarly, the denominator can be evaluated as

$$\begin{aligned}
\det(\mathbf{B}) &= \begin{vmatrix} (\gamma_0 + 1) & -c_1\gamma_1 & -c_2\gamma_2 & \dots & -c_N\gamma_N \\ -\gamma_1 & (\gamma_1 + 1) & 0 & \dots & 0 \\ -\gamma_2 & 0 & (\gamma_2 + 1) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\gamma_N & 0 & 0 & \dots & (\gamma_N + 1) \end{vmatrix} \\
&= \left\{ \prod_{j=1}^N [\gamma_j + 1] \right\} \left\{ (\gamma_0 + 1) - \sum_{j=1}^N \left[\frac{c_j \gamma_j^2}{(\gamma_j + 1)} \right] \right\} \\
&= \left\{ \prod_{j=1}^N [\gamma_j + 1] \right\} \left\{ \sum_{j=1}^N [c_j \gamma_j] + 1 - \sum_{j=1}^N \left[\frac{c_j \gamma_j^2}{(\gamma_j + 1)} \right] \right\} \\
&= \left\{ \prod_{j=1}^N [\gamma_j + 1] \right\} \left\{ 1 + \sum_{j=1}^N \left[\frac{c_j \gamma_j}{(\gamma_j + 1)} \right] \right\}
\end{aligned}$$

Thus, the solution for the zeroth entry in \mathbf{X} is

$$x_0^{n+1} = \frac{\det(\mathbf{B}_0)}{\det(\mathbf{B})} = \frac{x_0^n + S_2}{1 + S_1}$$

where

$$S_1 = \sum_{j=1}^N \left[\frac{c_j \gamma_j}{(\gamma_j + 1)} \right]$$

$$S_2 = \sum_{j=1}^N \left[\frac{c_j \gamma_j}{(\gamma_j + 1)} (x_j^n - a_j \tau_j) \right]$$

Once x_0^{n+1} is known, x_i^{n+1} can be determined from the original equations by back-substitution

$$\begin{aligned} x_i^{n+1} &= \frac{R_i + \gamma_i x_0^{n+1}}{(\gamma_i + 1)} \\ &= \frac{x_i^n + \gamma_i a_i \tau_i + \gamma_i x_0^{n+1}}{(\gamma_i + 1)} \\ &= \frac{x_i^n + \gamma_i (x_0^{n+1} + a_i \tau_i)}{(\gamma_i + 1)} \end{aligned}$$

Thus, the desired solution is obtained.

D.2 Bulk-Gas Properties

Since the multiphase model allows for multiple gas phases, each gas phase depends on its own material properties, has a distinct thermodynamic state, and resides in an isolated volume. However, actual gases would experience some mixing due to molecular diffusion. Therefore, the temperature- and velocity-relaxation models assume the gas phases mix instantaneously, resulting in a single combined bulk-gas phase where all the gases are in equilibrium. This mixing process results in a net increase in entropy.

In this section the equilibrium bulk-gas properties will be designated with the superscript *. The bulk-gas properties are generally computed by summing over the individual gas phases, indicated by the subscript i . The bulk-gas mass fraction x^* and volume fraction α^* are summations over the gas phases

$$x^* = \sum_i x_i$$

$$\alpha^* = \sum_i \alpha_i$$

The bulk density ρ^* , sound speed c^* , and thermal conductivity k^* are computed with a volume-weighted algebraic average

$$\begin{aligned}\rho^* &= \frac{\sum_i \alpha_i \rho_i}{\alpha^*} \\ c^* &= \frac{\sum_i \alpha_i c_i}{\alpha^*} \\ k^* &= \frac{\sum_i \alpha_i k_i}{\alpha^*}\end{aligned}$$

The bulk specific heats (C_v^* and C_p^*) use a mass-weighted algebraic average

$$\begin{aligned}C_v^* &= \frac{\sum_i x_i C_{vi}}{x^*} \\ C_p^* &= \frac{\sum_i x_i C_{pi}}{x^*}\end{aligned}$$

The bulk-gas viscosity μ^* uses a mass-weighted geometric average

$$\mu^* = \exp\left(\frac{\sum_i x_i \ln(\mu_i)}{x^*}\right)$$

The heat-transfer model (thermal relaxation) assumes the gases are in thermal equilibrium. Assuming that each gas phase enters the relaxation step with distinct temperatures, the equilibrium temperature is determined by applying a change in energy to each gas phase. Summing over all gas phases and applying the law of energy conservation gives

$$\sum_i x_i C_{vi} (T^* - T_i) = 0$$

Solving for the equilibrium temperature gives

$$T^* = \frac{\sum_i x_i C_{vi} T_i}{x^* C_v^*}$$

The thermal-relaxation model then enforces this temperature by updating the internal energy, ε

$$\Delta\varepsilon_i = C_{vi} (T^* - T_i)$$

Thus, the thermal-relaxation model assumes infinite heat conduction between the gases. The above equations assume constant thermal properties over the time-step.

Therefore, the thermal properties are evaluated at the beginning of the time-step in the above equations.

The bulk velocity \vec{u}^* is computed by conserving momentum. Summing over the change in momentum for each gas phase gives

$$\sum_i x_i (\vec{u}^* - \vec{u}_i) = 0$$

resulting in a mass-weighted algebraic average

$$\vec{u}^* = \frac{\sum_i x_i \vec{u}_i}{x^*} \quad (\text{D.9})$$

The temperature-relaxation model will compute this bulk velocity and use it to compute flow properties (such as Reynolds number), but will not actually modify the phase velocity. The velocity-relaxation model will actually modify the phase velocities such that

$$\vec{u}_i = \vec{u}^*$$

resulting in an infinite drag condition on the gas phases. This velocity update results in a change in the kinetic energy of the phase. Therefore, in order to conserve total energy in the zone, the total change in kinetic energy within the zone is redistributed into internal energy in the gas phases.

Appendix E

Evaluation of Discretization Error

Code verification plays an important role in software development. Roache [78] outlines systematic techniques for verifying numerical codes. These techniques require the evaluation of discretization errors. A measure of the discretization error is required to evaluate the order of accuracy for the numerical solution.

The discretization error for a quantity f at a particular location is defined as

$$E(\vec{x}, t) = f^{calc}(\vec{x}, t) - f^{exact}(\vec{x}, t)$$

where the point-wise error is the difference between the discrete solution, f^{calc} , and the exact (continuum) solution, f^{exact} . The above definition provides a local measure of the error at specific points, but it is often useful to quantify the error in a global sense. Several measures are possible for evaluation of the discretization error. The error metrics used in the current study are discussed in Salari and Knupp [79]. The current metrics utilize the L_2 -norm, given by

$$\|f^{calc} - f^{exact}\|_2 = \sqrt{\int_{\Omega} (f^{calc} - f^{exact})^2 d\mathcal{V}}$$

where the integrals are taken over the entire domain. Thus, the normalized global error is given by

$$E = \frac{\sqrt{\int_{\Omega} (f^{calc} - f^{exact})^2 d\mathcal{V}}}{\int_{\Omega} d\mathcal{V}}$$

Using single-point quadrature, the global error for a particular grid N can be computed as

$$E_N = \frac{\sqrt{\sum_{c=1}^N [(f_c^{calc} - f_c^{exact})^2 \Omega_c]}}{\sum_{c=1}^N [\Omega_c]}$$

where the sums are taken over each cell in the computational mesh and N is the total number of cells in the mesh. The subscript c indicates that the calculated and exact solutions are evaluated at the centroid of cell c . The global error is weighted with the cell volume, Ω_c , and it is normalized by the total volume of the domain. A simplification to the above metric is

$$E_N = \frac{1}{N} \sqrt{\sum_{c=1}^N [(f_c^{calc} - f_c^{exact})^2]} \quad (\text{E.1})$$

where the volume terms have been ignored. Salari and Knupp [79] show that (E.1) is sufficient for the purposes of code verification, thus it is the form used in the current study.

The error, given by (E.1) is a function of the mesh spacing, $E_N = E_N(\Delta x)$. For consistent numerical methods, the error follows

$$E_N(\Delta x) = C(\Delta x)^{r_N} + H.O.T.$$

where r_N is the order of accuracy for the method, C is an arbitrary constant, and $H.O.T.$ refers to higher-order terms. As the grid spacing, Δx , becomes smaller, the higher-order terms will vanish and the error will be dominated by the first term. Calculations can be run on a series of different sized grids, with the error evaluated on each grid. From this information, it is possible to acquire an observed order of accuracy for the calculations. If E_N^{grid1} and E_N^{grid2} are the global error evaluated on two meshes with mesh spacings of Δx^{grid1} and Δx^{grid2} , then the observed order of accuracy for the calculations is evaluated as

$$r_N = \frac{\ln\left(\frac{E_N^{grid1}}{E_N^{grid2}}\right)}{\ln\left(\frac{\Delta x^{grid1}}{\Delta x^{grid2}}\right)} \quad (\text{E.2})$$

Systematic grid refinement can then be done for additional grid levels, computing E_N and r_N at each level. The observed order of accuracy can then be compared to the theoretical order of the method.

REFERENCES

- [1] C. T. Crowe. Numerical-models for dilute gas-particle flows - review. *Journal Of Fluids Engineering-Transactions Of The Asme*, 104(3):297–303, 1982.
- [2] E. Loth. Numerical approaches for motion of dispersed particles, droplets and bubbles. *Progress In Energy And Combustion Science*, 26(3):161–223, 2000.
- [3] S. Elghobashi. On predicting particle-laden turbulent flows. *Applied Scientific Research*, 52(4):309–329, June 1994.
- [4] C. Crowe, M. Sommerfeld, and Y. Tsuji. *Multiphase Flows with Droplets and Particles*. CRC Press, 1998.
- [5] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal Of Computational Physics*, 152(2):457–492, July 1999.
- [6] R. Lohner, C. Yang, J. D. Baum, H. Luo, D. Pelessone, and C. M. Charman. The numerical simulation of strongly unsteady flow with hundreds of moving bodies. *International Journal For Numerical Methods In Fluids*, 31(1):113–+, September 1999.
- [7] M. Kucharik, R. V. Garimella, S. P. Schofield, and M. J. Shashkov. A comparative study of interface reconstruction methods for multi-material ALE simulations. *Journal Of Computational Physics*, 229(7):2432–2452, April 2010.
- [8] R. Ishii, Y. Umeda, and M. Yuhi. Numerical-analysis of gas particle 2-phase flows. *Journal Of Fluid Mechanics*, 203:475–515, June 1989.
- [9] S. Sivier, E. Loth, J. Baum, and R. Lohner. Dusty shock flow with unstructured adaptive finite elements and parcels. *AIAA Journal*, 34(5):1078–1080, May 1996.
- [10] A. Haselbacher, F. M. Najjar, and J. P. Ferry. An efficient and robust particle-localization algorithm for unstructured grids. *Journal Of Computational Physics*, 225(2):2198–2213, August 2007.
- [11] K. Balakrishnan, D. V. Nance, and S. Menon. Simulation of impulse effects from explosive charges containing metal particles. *Shock Waves*, 20(3):217–239, June 2010.
- [12] S. Sivier, E. Loth, J. Baum, and R. Lohner. Unstructured adaptive remeshing finite element method for dusty shock flow. *Shock Waves*, 4(1):15–23, July 1994.

- [13] E. Loth, S. Sivier, and J. Baum. Dusty detonation simulations with adaptive unstructured finite elements. *AIAA Journal*, 35(6):1018–1024, June 1997.
- [14] E. Loth, S. Sivier, and J. Baum. Adaptive unstructured finite element method for two-dimensional detonation simulations. *Shock Waves*, 8(1):47–53, February 1998.
- [15] M. V. Papalexandris. Numerical simulation of detonations in mixtures of gases and solid particles. *Journal Of Fluid Mechanics*, 507:95–142, May 2004.
- [16] J. M. Powers, D. S. Stewart, and H. Krier. Theory of 2-phase detonation .1. modeling. *Combustion And Flame*, 80(3-4):264–279, June 1990.
- [17] J. M. Powers, D. S. Stewart, and H. Krier. Theory of 2-phase detonation .2. structure. *Combustion And Flame*, 80(3-4):280–303, June 1990.
- [18] M. R. Baer and J. W. Nunziato. A 2-phase mixture theory for the deflagration-to-detonation transition (DDT) in reactive antigranulocytes-materials. *International Journal Of Multiphase Flow*, 12(6):861–889, November 1986.
- [19] R. Saurel and R. Abgrall. A multiphase godunov method for compressible multfluid and multiphase flows. *Journal Of Computational Physics*, 150(2):425–467, April 1999.
- [20] R. Abgrall and R. Saurel. Discrete equations for physical and numerical compressible multiphase mixtures. *Journal Of Computational Physics*, 186(2):361–396, April 2003.
- [21] D. A. Drew and S. L. Passman. *Theory of Multicomponent Fluids, Applied Mathematical Sciences, Vol. 135*. Springer, 1998.
- [22] A. Chinnayya, E. Daniel, and R. Saurel. Modelling detonation waves in heterogeneous energetic materials. *Journal Of Computational Physics*, 196(2):490–538, May 2004.
- [23] D. J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Computer Methods In Applied Mechanics And Engineering*, 99(2-3):235–394, September 1992.
- [24] D. J. Benson. Momentum advection on a staggered mesh. *Journal Of Computational Physics*, 100(1):143–162, May 1992.
- [25] E. J. Caramana, D. E. Burton, M. J. Shashkov, and P. P. Whalen. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *Journal Of Computational Physics*, 146(1):227–262, October 1998.

- [26] R. B. Pember and R. W. Anderson. Comparison of staggered-mesh lagrange plus remap and cell-centered direct Eulerian Godunov schemes for Eulerian shock hydrodynamics. Technical Report UCRL-JC-139820, Lawrence Livermore National Laboratory, 2000.
- [27] K. P. Ruggirello, P. E. DesJardin, M. R. Baer, M. J. Kaneshige, and E. S. Hertel. A reaction progress variable modeling approach for non-ideal explosives. In Preparation, *Journal of Fluid Mechanics*, 2011.
- [28] W. D. Henshaw and D. W. Schwendeman. Moving overlapping grids with adaptive mesh refinement for high-speed reactive and non-reactive flow. *Journal Of Computational Physics*, 216(2):744–779, August 2006.
- [29] C. Farhat, P. Geuzaine, and C. Grandmont. The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. *Journal Of Computational Physics*, 174(2):669–694, December 2001.
- [30] R. W. Smith. AUSM(ALE): A geometrically conservative arbitrary Lagrangian-Eulerian flux splitting scheme. *Journal Of Computational Physics*, 150(1):268–286, March 1999.
- [31] M. Lesoinne and C. Farhat. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Computer Methods In Applied Mechanics And Engineering*, 134(1-2):71–90, July 1996.
- [32] P. D. Thomas and C. K. Lombard. Geometric conservation law and its application to flow computations on moving grids. *AIAA Journal*, 17(10):1030–1037, 1979.
- [33] M. Vinokur. An analysis of finite-difference and finite-volume formulations of conservation-laws. *Journal Of Computational Physics*, 81(1):1–52, March 1989.
- [34] H. Zhang, M. Reggio, J. Y. Trepanier, and R. Camarero. Discrete form of the GCL for moving meshes and its implementation in CFD schemes. *Computers & Fluids*, 22(1):9–23, January 1993.
- [35] H. Luo, J. D. Baum, and R. Lohner. On the computation of multi-material flows using ALE formulation. *Journal Of Computational Physics*, 194(1):304–328, February 2004.
- [36] R. Saurel, J. Massoni, and F. Renaud. A numerical method for one-dimensional compressible multiphase flows on moving meshes. *International Journal For Numerical Methods In Fluids*, 54(12):1425–1450, August 2007.

- [37] J. Giordano, G. Jourdan, Y. Burtschell, M. Medale, D. E. Zeitoun, and L. Houas. Shock wave impacts on deforming panel, an application of fluid-structure interaction. *Shock Waves*, 14(1-2):103–110, June 2005.
- [38] D. L. Frost, C. Ornthanalai, Z. Zarei, V. Tanguay, and F. Zhang. Particle momentum effects from the detonation of heterogeneous explosives. *Journal Of Applied Physics*, 101(11):113529, June 2007.
- [39] J. Knap and D. E. Stevens. The application of multi-phase flow models in simulations of fluid-structure interaction. Technical Report UCRL-ABS-229402, Lawrence Livermore National Laboratory, 2007. Presented at Numerical methods for multi-material fluid flows, Prague, Czech Republic.
- [40] . . . Technical Report UCRL-SM-404490, Lawrence Livermore National Laboratory, Livermore CA, .
- [41] F. M. White. *Viscous Fluid Flow*. McGraw-Hill, Inc., 2nd edition, 1991.
- [42] F. M. White. *Fluid Mechanics*. McGraw-Hill, 4th edition, 1999.
- [43] K. K. Kuo. *Principles of Combustion*. John Wiley and Sons, Inc., 2nd edition, 2005.
- [44] G. H. Miller and P. Colella. A high-order Eulerian Godunov method for elastic-plastic flow in solids. *Journal Of Computational Physics*, 167(1):131–176, February 2001.
- [45] S. L. Gavriluk, N. Favrie, and R. Saurel. Modelling wave dynamics of compressible elastic materials. *Journal Of Computational Physics*, 227(5):2941–2969, February 2008.
- [46] V. A. Titarev, E. Romenski, and E. F. Toro. MUSTA-type upwind fluxes for non-linear elasticity. *International Journal For Numerical Methods In Engineering*, 73(7):897–926, February 2008.
- [47] A Zein. *Numerical methods for multiphase mixture conservation laws with phase transition*. PhD thesis, Magdeburg, Univ., 2010.
- [48] D. W. Schwendeman, C. W. Wahle, and A. K. Kapila. The Riemann problem and a high-resolution Godunov method for a model of compressible two-phase flow. *Journal Of Computational Physics*, 212(2):490–526, March 2006.
- [49] S. A. Tokareva and E. F. Toro. HLLC-type riemann solver for the Baer-Nunziato equations of compressible two-phase flow. *Journal Of Computational Physics*, 229(10):3573–3604, May 2010.
- [50] R. Aris. *Vectors, Tensors, and the Basic Equations of Fluid Mechanics*. Prentice-Hall, 1962.

- [51] A. Guardone and L. Vigevano. Finite element/volume solution to axisymmetric conservation laws. *Journal Of Computational Physics*, 224(2):489–518, June 2007.
- [52] C. Hirsch. *Numerical Computation of Internal and External Flows. Volume 1. Fundamentals of Numerical Discretization*. John Wiley & Sons, 1st edition, 2001.
- [53] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics, A Practical Introduction*. Springer, 2nd edition, 1999.
- [54] G. Strang. On construction and comparison of difference schemes. *Siam Journal On Numerical Analysis*, 5(3):506–517, 1968.
- [55] O. Y. Vorobiev, I. N. Lomov, A. V. Shutov, V. I. Kondaurov, A. L. Ni, and V. E. Fortov. Application of schemes on moving grids for numerical simulation of hypervelocity impact problems. *International Journal Of Impact Engineering*, 17(4-6):891–902, 1995.
- [56] S. K. Godunov. A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics. *Mat. Sb*, 47(3):357–393, 1959.
- [57] G. H. Miller and E. G. Puckett. A high-order Godunov method for multiple condensed phases. *Journal Of Computational Physics*, 128(1):134–164, October 1996.
- [58] M. Sun and K. Takayama. An artificially upstream flux vector splitting scheme for the Euler equations. *Journal Of Computational Physics*, 189(1):305–329, July 2003.
- [59] J. Knap, D. E. Stevens, T. A. Dunn, and J. R. Kercher. Multi-phase modeling of solids. Submitted to *Journal of Computational Physics*, 2008.
- [60] K. W. Thompson. Time-dependent boundary-conditions for hyperbolic systems. *Journal Of Computational Physics*, 68(1):1–24, January 1987.
- [61] K. W. Thompson. Time-dependent boundary-conditions for hyperbolic systems .2. *Journal Of Computational Physics*, 89(2):439–461, August 1990.
- [62] N. Balakrishnan and G. Fernandez. Wall boundary conditions for inviscid compressible flows on unstructured meshes. *International Journal For Numerical Methods In Fluids*, 28(10):1481–1501, December 1998.
- [63] R. A. Berry, R. Saurel, and O. LeMetayer. The discrete equation method (DEM) for fully compressible, two-phase flows in ducts of spatially varying cross-section. *Nuclear Engineering And Design*, 240(11):3797–3818, November 2010.

- [64] M. S. Liou. A sequel to AUSM: AUSM(+). *Journal Of Computational Physics*, 129(2):364–382, December 1996.
- [65] M. Berger, M. J. Aftosmis, and S. M. Murman. Analysis of slope limiters on irregular grids. In *AIAA Paper 2005-0490*, number 2005-0490. American Institute of Aeronautics and Astronautics, 2005.
- [66] T. Barth. Numerical methods for conservation laws on structured and unstructured meshes. VKI Lecture Series, March 2003.
- [67] M. H. Lallemand, A. Chinnayya, and O. Le Metayer. Pressure relaxation procedures for multiphase compressible flows. *International Journal For Numerical Methods In Fluids*, 49(1):1–56, September 2005.
- [68] R. Tipton. CALE users manual. Technical report, Lawrence Livermore National Laboratory, Livermore, CA, 1990.
- [69] J. B. Bdzil, R. Menikoff, S. F. Son, A. K. Kapila, and D. S. Stewart. Two-phase modeling of deflagration-to-detonation transition in granular materials: A critical examination of modeling issues. *Physics Of Fluids*, 11(2):378–402, February 1999.
- [70] R. Saurel, N. Favrie, F. Petitpas, M. H. Lallemand, and S. L. Gavriluk. Modelling dynamic and irreversible powder compaction. *Journal Of Fluid Mechanics*, 664:348–396, December 2010.
- [71] K. P. Ruggirello, P. E. DesJardin, M. R. Baer, and E. S. Hertel. Modeling of particle compressibility and ignition from shock focusing. *Combustion Theory And Modelling*, 14(1):41–67, 2010.
- [72] T. Belytschko, W. K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. Wiley, 2000.
- [73] G. L. Goudreau and J. O. Hallquist. Recent developments in large-scale finite-element lagrangian hydrocode technology. *Computer Methods In Applied Mechanics And Engineering*, 33(1-3):725–757, 1982.
- [74] M. L. Wilkins. Calculation of elastic plastic flow. Technical Report UCRL-7322, University of California, Lawrence Radiatin Laboratory, Livermore, California, 1963.
- [75] P. H. Maire, R. Abgrall, J. Breil, and J. Ovadia. A cell-centered Lagrangian scheme for two-dimensional compressible flow problems. *Siam Journal On Scientific Computing*, 29(4):1781–1824, 2007.
- [76] A. M. Winslow. Equipotential zoning of two dimensional meshes. Technical Report UCRL-7312, Lawrence Radiation Laboratory, 1963.

- [77] B. I. Jun. Modified equipotential method for grid relaxation. Technical Report UCRL-JC-138277, Lawrence Livermore National Laboratory, 2000.
- [78] P. J. Roache. Code verification by the method of manufactured solutions. *Journal Of Fluids Engineering-Transactions Of The Asme*, 124(1):4–10, March 2002.
- [79] K. Salari and P. Knupp. Code verification by the method of manufactured solutions. Technical Report SAND2000-1444, Sandia National Labs., Albuquerque, NM (United States), 2000.
- [80] G. A. Sod. Survey of several finite-difference methods for systems of non-linear hyperbolic conservation laws. *Journal Of Computational Physics*, 27(1):1–31, 1978.
- [81] P. Woodward and P. Colella. The numerical-simulation of two-dimensional fluid-flow with strong shocks. *Journal Of Computational Physics*, 54(1):115–173, 1984.
- [82] J. D. Anderson, Jr. *Modern Compressible Flow With Historical Perspective*. McGraw-Hill, 1982.
- [83] F. X. Timmes, G. Gisler, and G. M. Hrbek. Automated analyses of the tri-lab verification test suite on uniform and adaptive grids for code project a. Technical Report LA-UR-05-6865, Los Alamos National Laboratory, 2005.
- [84] L. I. Sedov. *Similarity and Dimensional Methods in Mechanics*. Academic Press, 1959.
- [85] W. F. Noh. Errors for calculations of strong shocks using an artificial viscosity and an artificial heat-flux. *Journal Of Computational Physics*, 72(1):78–120, September 1987.
- [86] R. Saurel, F. Petitpas, and R. A. Berry. Simple and efficient relaxation methods for interfaces separating compressible fluids, cavitating flows and shocks in multiphase mixtures. *Journal Of Computational Physics*, 228(5):1678–1712, March 2009.
- [87] M. J. Ivings, D. M. Causon, and E. F. Toro. On Riemann solvers for compressible liquids. *International Journal For Numerical Methods In Fluids*, 28(3):395–418, September 1998.
- [88] J. W. Banks, D. W. Schwendeman, A. K. Kapila, and W. D. Henshaw. A high-resolution Godunov method for compressible multi-material flow on overlapping grids. *Journal Of Computational Physics*, 223(1):262–297, April 2007.
- [89] R. D. Zucker. *Fundamentals of Gas Dynamics*. Matrix Publishers, Inc., 1977.

- [90] Y. B. Zel'dovich and Y. P. Raizer. *Physics of Shock Waves and High-Temperature Hydrodynamic Phenomena*. Dover, 2002.
- [91] . *SciPy Reference Guide*. SciPy Community, release 0.10.0.dev edition, 2011.
- [92] M. Sommerfeld. The unsteadiness of shock-waves propagating through gas-particle mixtures. *Experiments In Fluids*, 3(4):197–206, 1985.
- [93] J. H. Mathews. *Numerical Methods For Mathematics, Science, and Engineering*. Prentice Hall, 2nd edition, 1992.
- [94] B. Larrouturou. How to preserve the mass fractions positivity when computing compressible multicomponent flows. *Journal Of Computational Physics*, 95(1):59–84, July 1991.
- [95] P. H. Maire. A high-order cell-centered Lagrangian scheme for compressible fluid flows in two-dimensional cylindrical geometry. *Journal Of Computational Physics*, 228(18):6882–6915, October 2009.
- [96] E. Daniel, R. Saurel, and J. C. Loraud. Modeling of axisymmetric two-phase dilute flows. *International Journal Of Computational Fluid Dynamics*, 10(4):301–313, 1998.
- [97] Ralph Menikoff. Empirical equations of state for solids. In Yasuyuki Horie, editor, *ShockWave Science and Technology Reference Library*, pages 143–188. Springer Berlin Heidelberg, 2007.
- [98] O. Le Metayer, J. Massoni, and R. Saurel. Elaborating equations of state of a liquid and its vapor for two-phase flow models. *International Journal Of Thermal Sciences*, 43(3):265–276, March 2004. (*In French*).
- [99] C. E. Castro and E. F. Toro. A Riemann solver and upwind methods for a two-phase flow model in non-conservative form. *International Journal For Numerical Methods In Fluids*, 50(3):275–307, January 2006.
- [100] B. J. Plohr. Shockless acceleration of thin plates modeled by a tracked random choice method. *AIAA Journal*, 26(4):470–478, April 1988.
- [101] J. P. Cocchi and R. Saurel. A Riemann problem based method for the resolution of compressible multimaterial flows. *Journal Of Computational Physics*, 137(2):265–298, November 1997.
- [102] B. M. Dobratz and P. C. Crawford. LLNL explosives handbook, properties of chemical explosives and explosive simulants. Technical Report UCRL-52997, Lawrence Livermore National Laboratory, University of California, Livermore, CA 94550, 1985.
- [103] D. Gidaspow. *Multiphase Flow and Fluidization, Continuum and Kinetic Theory Descriptions*. Academic Press, 1994.

- [104] J Lundberg and B Halvorsen. A review of some existing drag models describing the interaction between phases in a bubbling fluidized bed. In *49th Scandinavian Conference on Simulation and Modeling*, 2008. SIMS 2008, Oslo University college Oct 7-8, 2008-10-08.
- [105] I. S. Akhatov and P. B. Vainshtein. Transition of porous explosive combustion into detonation. *Combustion Explosion And Shock Waves*, 20(1):63–69, 1984.
- [106] E. Angerson, Z. Bai, J. Dongarra, A. Greenbaum, A. McKenney, J. Du Croz, S. Hammarling, J. Demmel, C. Bischof, and D. Sorensen. LAPACK: A portable linear algebra library for high-performance computers. In *Supercomputing '90. Proceedings of DOI - 10.1109/SUPERC.1990.129995*, pages 2–11, 1990.
- [107] H. Anton. *Elementary Linear Algebra*. Wiley, 5th edition, 1987.